MODELING OF MONTHLY RUNOFF TIME-SERIES USING ARTIFICIAL NEURAL NETWORKS

A Thesis Submitted
In partial Fulfillment of the Requirements
for the degree of
MASTER OF TECHNOLOGY

BY
MADHAV KUMAR.A

to the

Department of Civil Engineering
INDIAN INSTITUTE OF TECHNOLOGY KANPUR

NOVEMBER, 2000

MODELING OF MONTHLY RUNOFF TIME-SERIES USING ARTIFICIAL NEURAL NETWORKS

BY

MADHAV KUMAR.A

Department of Civil Engineering

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

NOVEMBER, 2000

CERTIFICATE

It is certified that the work contained in the thesis entitled " MODELING OF MONTHLY RUNOFF TIME-SERIES USING ARTIFICIAL NEURAL NETWORKS" , by MADHAV KUMAR.A (Roll No.9820303), has been carried out under my supervision and this work has not been submitted elsewhere for a degree.

(Dr. Ashu Jain)

Assistant Professor
Department of Civil Engineering
Indian Institute of Technology
Kanpur, INDIA

November, 2000

Dedicated to

My Grand Mother

and

My Beloved Wife Pramee

Abstract

Two types of modeling approaches have been investigated to model monthly runoff data. The first type of models uses the time-series technique of autoregressive modeling, while the second type of models employ a relatively new technique of Artificial Neural Networks (ANNs). The ANN technique was applied in a time-series mode in which ANN models were developed using raw data, detrended data and detrended and deseasonalized data. The monthly runoff data from the Colorado River at Lees Ferry U.S.A., for a period of 62 years were employed in this study. For all models developed in this study, the data of 57 years were used for the calibration purposes, and remaining data were used to test the performance of the models using certain standard statistical parameters. It has been found that the ANN models provide a better representation of runoff prediction as compared to the AR models.

Acknowledgements

I feel highly indebted to my Thesis Supervisor Dr. Ashu Jain for his most efficient and invaluable guidance at every stage in my thesis work. His valuable suggestions and encouragement inspired me to take up the task challengingly. I am enriched with the practical side as well as theoretical side with the abundance of analytical and practical techniques possessed by the guide. His extraordinary patience benefited me to clear doubts all over the study. I had a great experience in my life by completing this thesis successfully with my guide. Ultimately it was a great moment to work under his supervision.

I thank gratefully Prof. T.Gangadharaiah for his valuable suggestions and advices during my M.tech programme. I am also thankful to Dr. Bithin Dutta and Dr. Rajesh Srivastava for their kind cooperation during my thesis work.

I am lucky to get nice friends at IIT Kanpur. Their encouragement, love and support made me to complete my thesis successfully. My seniors also gave me good cooperation. Special thanks goes to my friends N.Sreenivas, A.Ramesh, R.Aravind, K.Manjula, Shabana and Ritesh. I enjoyed well with them in partying and in picnics.

I am greatly indebted to Prameela, my wife, for her prudent cooperation and excellent guidance helped me to complete this thesis successfully. I was always free from tensions with her company. She played a vital role in completing M.Tech programme.

My deep sense of gratitudes to my parents with their encouragement to do postgraduate studies. I specially thanks to them

My brothers Ramakrishna and Kiran Babu gave good support at some key stages of my M.Tech programme. I enjoyed well with them in partying.

I also express my thanks to Ph.D scholars Sudhakar, Sreenivasulu and Prasad.

I finally express thanks to all my friends who encouraged me directly or indirectly to complete my thesis successfully.

November, 2000

A.Madhav Kumar
Indian Institute of Technology
Kanpur

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General

Runoff forecast are useful in design and control of water resources systems such as water supply, reservoir operation, flood and drought management *etc.* Hydrologic simulations of watershed based on physical and mathematical concepts have been the focus of attention of researchers since 1960's. Recent developments in computers and analysis techniques have led to significant developments and applications of mathematical and conceptual models in hydrology. The model existence differs in the inter-relationship between their various components and their computational time steps. One of the first steps in runoff modeling is to identify the kind of model that is suitable for a data set from a particular watershed within the limited amount of resources available.

1.2 Models for Runoff process

Considering various aspects of hydrological investigations, the hydrologic models can be classified into three broad categories: deterministic models, conceptual models and stochastic or black-box models. Deterministic models are formulated by the set of variables affecting the rainfall-runoff process and parameters and equations relating to them. They are complicated and computationally expensive, so normally conceptual models are developed. Conceptual models are formulated on the basis of a simple arrangement of a relatively small number of components, each of which is simplified

representation of one process element in the system being modeled. Most of the conceptual models are lumped representation of parameters. The third modeling approach is called stochastic or black box approach. A system is stochastic if its behavior is governed by laws of probability. A black-box model uses an appropriate mathematical function which is fitted to the data without considering the physical process it represents. Black-box models are easy to develop and implement. Time-series models for runoff forecasting fall under this category. Since the early nineties, Artificial Neural Networks (ANNs) have been successfully used in hydrology-related areas such as runoff modeling, precipitation forecasting, hydrologic time series prediction *etc.* An ANN is a massively parallel-distributed information processing system that has certain performance characteristics resembling biological neural network of the human brain. ANN would have to be classified as black-box models, as they do not consider the physical process underlying the phenomenon being modelled.

ANNs have been applied in a wide variety of areas in engineering. Cheng Yeh (1999) used neural networks to design of High-Performance concrete mixture in structural engineering. Abdullah and Ali (1998) applied ANN approach for pavement maintenance. Wand, *et el.* (1997) used ANN to prediction of pile capacity in geotechnical engineering. Pezeshk and Camp (1996) used neural networks to geographical log interpretation. Anthony and Goh (1995) applied ANN to modeling soil correlations. Markus, Salas and Shin (1995) used neural networks to predicting stream flows. Karunanidhi, *et el.* (1994) applied ANN to river flow prediction.

A typical modeling application consists of the following steps i) selecting the type of model based on study objectives and characteristics of the system to be studied ii) deciding the structure of the model to be developed iii) calibrate model using calibration data set to identify the model parameters for

a particular application. iv) Validate model using validation data set. v) Apply validated model in the forecasting.

1.3 Objectives of the Present Study

The primary objective of this thesis is to develop mathematical models of black-box type for short term runoff forecasting. Both time-series analysis and ANN technique will be explored for this purpose. First time-series models of auto-regressive (AR) type will be developed. Then the ANN technique will be applied for forecasting monthly runoff. While developing ANN models for monthly runoff forecasting, the ANN technique will be applied to different time-series of runoff. These include: a) original runoff time-series b) detrended runoff time-series c) detrended and deseasonalised runoff time-series. Various types of ANN architectures will be explored for this purpose using various time-series in order to arrive at the best ANN model.

The development of time-series and ANN models will require the following steps to be carried out.

1) Obtain monthly data for sufficient length, and break up data into calibration/training and validation/testing sets.

2) Develop separate computer codes for modeling and forecasting runoff using both time-series and ANN techniques. Check the correctness of the computer program using some hypothetical and real data sets.

3) Develop runoff models by determining parameters of models using calibration/training data set and test their performance using validation/testing data set.

1.4 Organization of the Thesis

This Chapter discusses runoff modeling in general, modeling techniques available and the objectives and organization of the thesis. Chapter 2 reviews the literature available in the area of runoff modeling. An introduction to the relatively new technique of ANNs is presented in chapter 3. Chapter 4, discusses autoregressive models. Results and discussions are presented in Chapter 5, while concluding remarks are made in Chapter 6. References and appendices are provided at the end.

Chapter 2

Literature Review

Many models have been developed by various researchers for runoff modeling. This chapter provides a brief description of various models such as deterministic models, conceptual models, stochastic models such as time-series models, and the relatively new Artificial Neural Network (ANN) models.

The most comprehensive watershed model, called Stanford Watershed Model 4 (SWM4), was developed for river flow modeling and forecasting (Linsley 1964). The operation of this model is controlled by 30 parameters. The detailed parameter listing and operational specifications of this model are available in Fleming (1975). The incoming rainfall either becomes direct runoff or is detained in upper and lower soil moisture storages. The three storage zones combine to represent the effects of highly variable soil moisture profiles and ground water contribution. The upper zone storage absorbs a large part of the first few hours of rain in a storm. The lower zone storage controls long-term infiltration. The ground water storage controls base flow in the stream. The direct runoff is split into two components, surface runoff and interflow. Total river flow is the sum of surface runoff, interflow and base flow. To apply the model on a split-test basis, the typical procedure is to select some portion of rainfall and runoff records for a catchment. This period is used to develop estimates of the model parameters that fit the general model to the given catchment. A second period of record is then used as a control to check the accuracy of the parameters obtained from the first period. A model of the complexity such as that of SWM4 requires skill,

experience and judgment from its operator in making the parameter adjustments needed for acceptable fitting.

Another example of deterministic model for runoff forecast is the Dawdy-O'Donnel model, (O'Donnel 1965). The O'Donnel model has four storage elements: a surface storage, a channel storage, a soil moisture storage and a ground water storage. This model has nine parameters. When calibrating this type of model, the use of records beginning with a long, dry period is recommended so that the four storage elements can be allocated to zero values and the potential infiltration rate set to its maximum. The nine parameter values must then be determined from records of rainfall, stream flow and potential evaporation, using either trial-and-error methods or automatic optimization procedures. This model requires a subtle combination of experience and intuition, since obviously the temporal variations of the output stream flow are more sensitive to some parameters.

Another model called the Sacramento Soil Moisture Accounting (SAC-SMA) model was developed by Burnesh *et al.* (1973) mainly for flood forecasting purposes. The inputs to the SAC-SMA model are precipitation and evaporatranspiration. Precipitation is provided in the form of a mean areal precipitation (average precipitation over the entire soil moisture accounting area). The outputs from the model are estimated evapotranspiration and channel flow; the latter is converted into stream flow by means of a unit hydrograph. The SAC-SMA model has 16 parameters. One of the global optimization method, the Shuffled Complex Evaluation (SCE-UA) method is able to find the optimal parameter set during calibration of the SAC-SMA. Due to its complex structure, the SAC-SMA model has not gained much popularity.

Recently, the soil moisture module of the ARNO model has been extensively used in hydrological practice, particularly for flood forecasting purposes. The model, which derives its name from its first application to the Arno River. It was developed by the Commission of the European Communities (European Flood Forecasting Operational System 1992). In the ARNO model, the linear parabolic approach has been successfully used with the parameter values that can be established according to physical reasoning, without the need of extensive trial and error or optimization procedures.

The models discussed above are rainfall-runoff models of either deterministic or conceptual type. They require runoff, rainfall and evapotranspiration data. The runoff models are useful when rainfall data are not available. The runoff models of black-box type reported in literature are discussed in brief here.

Autoregressive (AR) models have been extensively used in hydrology and water resources since 1960's, for modeling annual and periodic hydrologic time series. The application of these models has been attraction in hydrology mainly because (i) the autoregressive form has an intuitive type of time dependence (the value of a variable at the present time depends on the values at previous times), and (ii) they are simplest models to use.

Thomos and Fiering (1962) and Yevjevich (1973) were probably the first ones to develop AR models in hydrology. The usual procedure for estimating the parameters of the models has been based on method of moments and the test of goodness of fit of the model was based on the correlogram analysis.

Delleur and Kavvas (1978) applied ARMA models to the weekly and daily flow series over 15 basins located in Indiana, Illinois and Kentucky. Monthly flow of 16 watersheds located in these three states were later studied by McKerchar and Delleur (1974). They found that the ARIMA models require less parameters than

ARMA counterparts. However, the principal limitation of ARIMA models as compared to ARMA models was that the ARIMA models, in general, were not suitable for simulation (Watts 1972). Box and Jenkins (1976) used time series analysis for flood forecasting and control. Bolzern, *et al.* (1980) used time series analysis for adaptive real-time forecast of river flow rates from rainfall data. Burn and Mc Bean (1985) applied time series analysis for river flow forecasting model for Sturgeon River. Georgakakos (1989) used time series analysis to the values to the value of stream flow forecasting in reservoir operation. Restrepo, *et al.* (1992) used time series analysis for real time stream flow forecasting and control on the Hun River Basin, Korea.

The most widely used computer model for monthly river flow simulation is HEC-4. HEC-4 was developed by U.S. Army Corps of Engineers at the Hydrologic Engineering Center in 1971. Statistical characteristics used in the generation are calculated from observed monthly river flows. Missing data are calculated based on concurrent flows at other stations. Each monthly flow is converted to a normalized standard variate using the Pearson TypeIII approximation. Simple coefficients of correlation between all pairs of stations for each current and preceeding calendar month are computed for the normalized flows by using some equations. Hypothetical monthly river flow volumes are generated computing a regression equation by the Crout Method, for each station and month and then computing river flows for each station for one month at a time.

Recently, Artificial Neural Networks have been successfully applied to many application in hydrology. Markus *et al.* (1995) used ANNs with the back-propagation algorithm to predict monthly river flows at the Del Norte gauging station in the Rio Grande Basin in South Colorado. The results indicated that ANNs did a good job of predicting stream flows.

The neural network approach is applied to the flow prediction of the Hurm River at the Dextur sampling station, Mich (Karunanidhi, 1994). Empirically comparisons are performed between the predictive capabilities of the neural network models and the most commonly used analytic non linear power model in terms of accuracy and convenience of use. Preliminary results are quite encouraging.

Raman and Sunilkumar (1995) employed an ANN to model a multivariate water resources time series and compared with those obtained by traditional autoregressive moving average (ARMA) models. The objective was to synthesize monthly inflow data for two reservoir sites in Dharathapuzha basin in South India. They concluded that the results obtained using the ANN compared well with those obtained using statistical methods. And some more successful applications in runff simulation include Kang (1993) Karunanidhi *et al.* (1994), Poff *et al.* (1996), Muttiah *et al.* (1997), Tawfic *et al.* (1997), Thirumalaiah and Deo (1998).

Chapter 3

Artificial Neural Networks

3.1 General

Artificial neural networks (ANNs) are inspired by the structure of the human brain that is well suited for complicated tasks such as river flow modeling, precipitation forecasting *etc.*, in hydrologic systems (Taglisrini *et al.* 1991). There has been an increased interest in ANNs during recent years. The ANNs emerged after the introduction of simplified neurons by Mc Culloch and Pitts (Mc Culloch and Pitts 1943). These neurons were presented as models of biological neurons and as conceptual components that could perform computational tasks. ANNs have the ability to learn from examples and modify their behavior in response to their surrounding environment. ANNs have been proven to provide better solutions for simulation and forecasting. Before looking at the structure of an ANN, let us look at the structure of a biological neuron.

3.2 The Biological Neuron

The human brain is the most complex computing device known. The brain's powerful thinking of remembering, and problem solving capabilities inspired many scientists to attempt computer modeling of its operation. The brain of average human being consists of billions of neurons (10^{11}) which are densely interconnected. Each neuron is a micro-processing (see Figure 3.1) unit built up of three parts: the cell body, the dendrites, and the axon. As shown in Figure

3.1, the axon splits up and connects to dendrites of other neurons through functions referred to as synapses. A neuron receives and combines signals from other neurons through the dendrites and if the combined signal is strong enough, it causes the neuron to fire producing an output signal. The output signal travels along the axon to other receiving neurons. The magnitude of the signal sent depends on the amount of chemical released by the axon and received by the dendrites. The synaptic efficiency or "strength" is what is modified when the brain learns (Hebb 1949). The synapse combined with the processing of information in the neuron forms the basic memory mechanism of the brain.

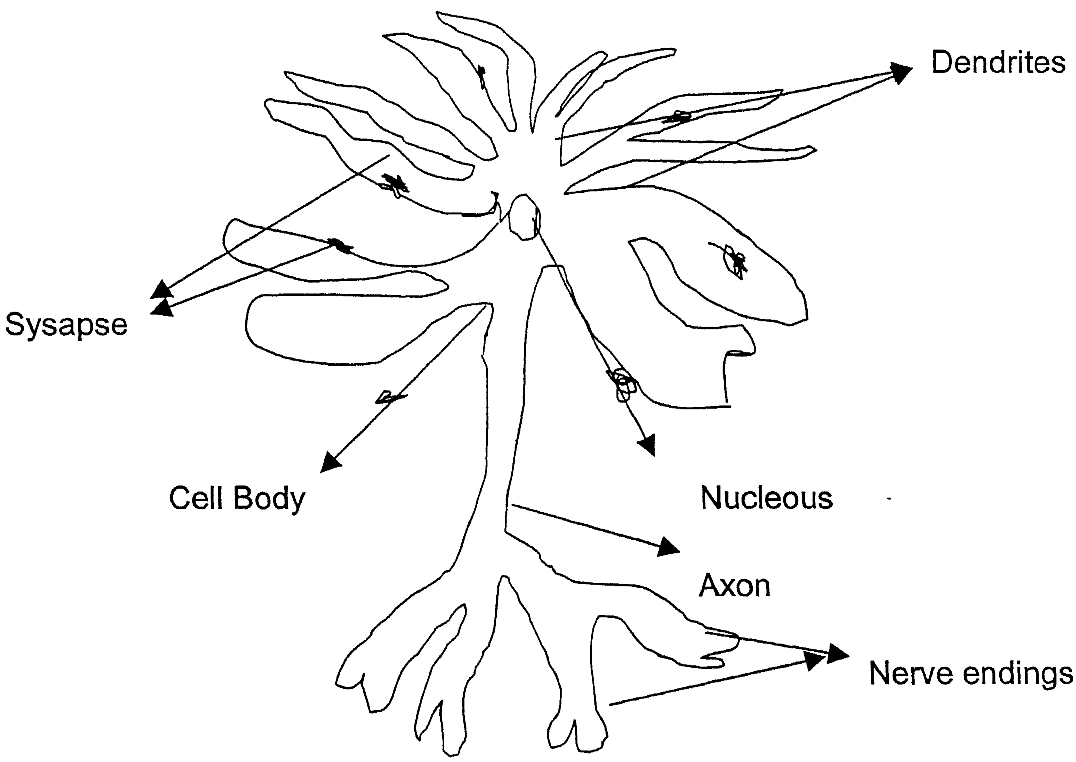


Figure 3.1: Structure of a Biological Neuron

3.3 Artificial Neural Networks

An ANN is an information processing system that is composed of a number of processing elements or artificial neurons analogous to biological neurons and inter connections or weights between these elements that imitate the synaptic strength in a biological nervous system. The ANN approach is based on the highly interconnected structure of the brain cells. This approach is faster compared with its conventional compatriots, robust in noisy environments, flexible in the range of problems it can solve, and highly adaptive to the newer environments. Due to these established advantages, currently the ANNs have numerous real world applications. Extensive research has been carried out on its implementation in the system engineering related fields such as time series prediction, river flow modeling, and rainfall-runoff modeling.

In order for an ANN to generate an output value that is as close as possible to the target value, a training process, also called learning is employed. The process of training is an important aspect, and the performance of an ANN is crucially dependent on successful training.

There are primarily two types of training; supervised and unsupervised. A supervised training algorithm requires an external teacher to guide the training process. This typically implies that a large number of examples (or patterns) of inputs and outputs are required for training. The inputs are cause variables of a system and the outputs are the effect variables. The training procedure involves the iterative adjustment of connection between weights and threshold values for each of the nodes. The primary goal of training is to minimize the error function by searching for a set of connection strengths and threshold values that cause the ANN to produce outputs that are equal or closer to targets.

ANNs are methods for empirically mapping inputs to outputs with no specification of the form of the relationship, which leaves them highly sensitive to the composition of the samples used to train them. The fact that different training samples produce different ANNs does not, however mean that the optimal solution sets will be sensitively changed.

3.4 ANN Architecture

A neural network is characterized by its architecture that represent the pattern of connection between nodes. The architecture of an ANN is designed by weights between neurons, a transfer function that controls the generation of output in a neuron, and learning laws that define the relative importance of weights for input to a neuron. The architecture of an ANN is classified into two types: single hidden layer ANN and multi hidden-layer ANN.

3.4.1 Single Hidden- Layer ANN

Neurons in an ANN are arranged in groups called layers or slabs. The nodes in one layer are connected to those in the next, but not to those in the same layer. ANNs can also be characterized based on the direction of information flow and processing. In a feed-forward network, the weighted connections feed activations only in the forward direction from the input layer to output layer. On the other hand, in a recurrent network additional weighted connections are used to feed previous activations back into the network.

A single hidden-layer ANN consists of one input layer, one hidden layer and one output layer. The structure of single hidden-layer ANN is shown in Figure 3.2.

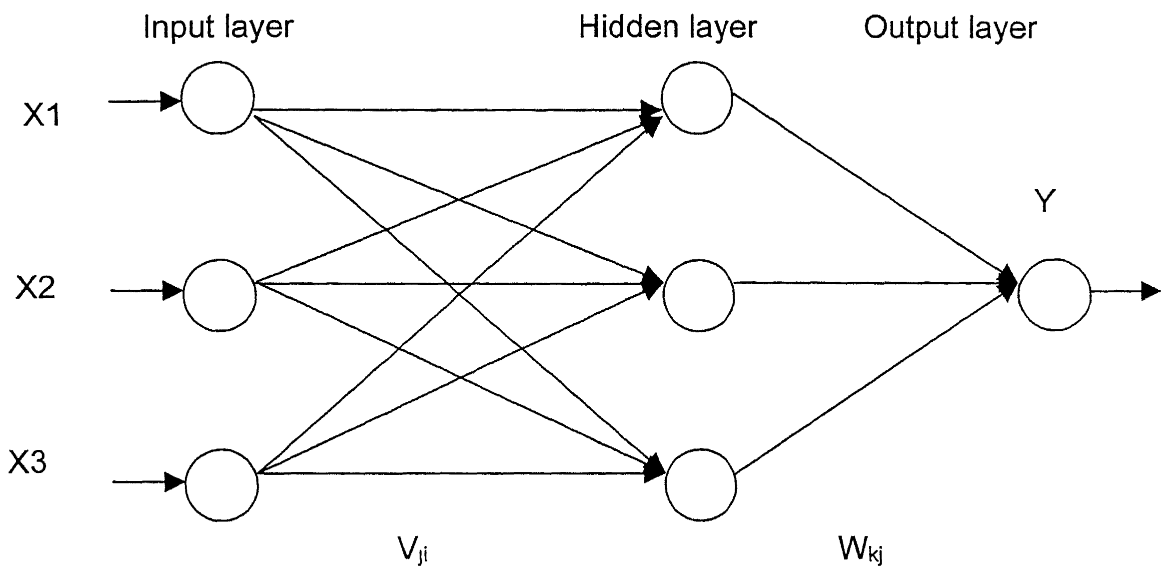


Figure 3.2: Single Hidden-Layer ANN

As shown in Figure 3.2, X1 X2 and X3 are inputs. Circles are the neurons. Each neuron simply computes output of a weighted sum of the inputs to the network. The connection between the neurons, represented by lines, is quantified by their weights, which are shown in the form V_{ji} and W_{kj}, Y is the output from the single hidden-layer ANN.

3.4.2 Multi Hidden-Layer ANN

Multi hidden-layer ANN is one of the most widely used classes of ANNs. Each such ANN consists of an input layer, an output layer and one or more intermediate, hidden layers as shown in Figure 3.3.

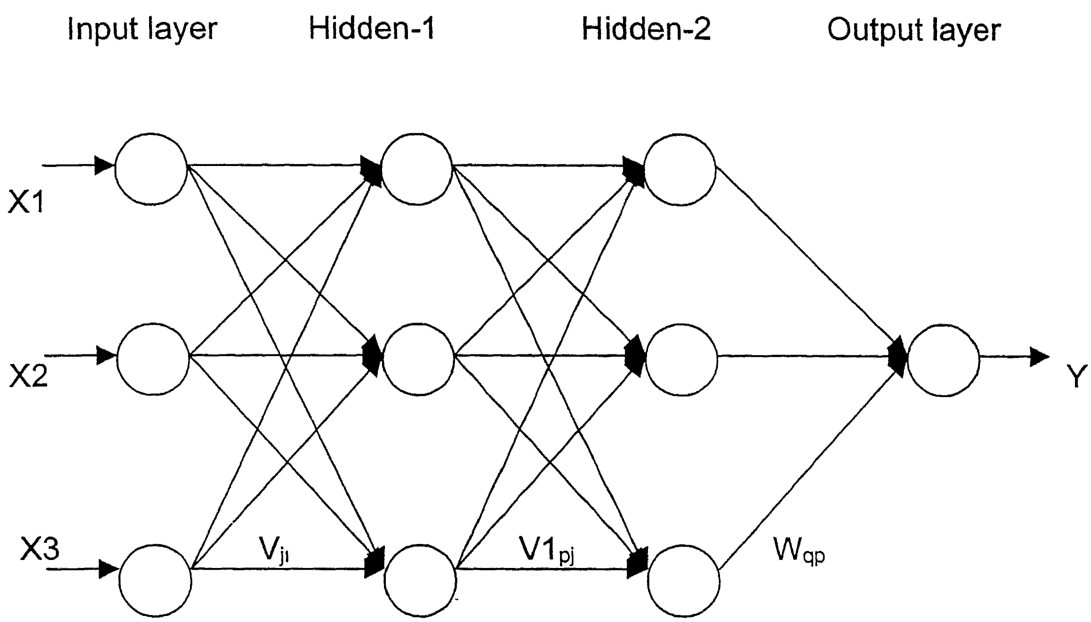


Fig 3.3: Multi Hidden-Layer ANN

Inputs are shown by X1, X2 and X3 and V_{ji} represents the connection weight from the j^{th} node in the preceding layer to i^{th} node. 'Y' is the observed output of the network. The most commonly used learning algorithm for multi-layer ANNs is the "back-propagation algorithm".

3.4.2.1 Back Propagation Training Algorithm

Back-propagation training algorithm is the most commonly used supervised algorithm for training the multi hidden-layer ANNs. An ANN which uses back-propagation algorithm for its training is also called back-propagation ANN. In back-propagation ANNs, information is processed in the forward direction from the input layer to the hidden layer(s) and then to output layer. The objective of a back-propagation network is to find the weights that approximate target values of output with a selected accuracy. The least-mean-square-error method, along with the generalized-delta rule, is used to optimize the network weights in back-

propagation networks. The gradient-descent method along with the chain rule of the derivative, is employed to modify the network weights. It requires a continuous, differentiable and non-linear function on the ANN to compute output from each neuron.

The input data are multiplied by the initial weights, then the weighted inputs are added by simple summation to yield the net input (say net) to each neuron.

$$Net = \sum_{i=1}^{N} V_{ji}\ X_i \qquad (3.1)$$

where X_i = input to any neuron

V_{ji} = weighted matrix from jth layer to Ith layer

N = number of inputs

Net = net for jth neuron

The net of neuron is passed through an activation or transfer function to produce output from a neuron

$$O = \frac{1}{1+\exp(-Net)} \qquad (3.2)$$

Where O = output signal from ith neuron

After the output of the neuron is transmitted to the next layer as an input, this procedure is repeated until the output layer is reached. This is called a forward pass.

The error between the output of the network and the target output are computed at the end of each forward pass, and is summed over as follows:

$$E = \sum_{i=1}^{N} \frac{1}{2}(O_i - D_i)^2 \qquad (3.3)$$

where E = Total Error

O_i = Observed output

D_i = Target output

The weight values are originally initialized randomly for all the connection weights in the network. During the back-propagation of error signal at output neuron, the weights are modified according to the following equations:

$$V_{ji}(n+1) = V_{ji}(n) + \Delta V_{ji}(n) \qquad (3.4)$$

$$\Delta V_{ji}(n) = \eta(\delta_i)(O_j) + \alpha\ \Delta V_{ji}(n-1) \qquad (3.5)$$

where

$\Delta V_{ji}(n)$	=	change in weight V_{ji} at n^{th} iteration
$\Delta V_{ji}(n-1)$	=	change in weight V_{ji} at $n-1^{th}$ iteration
$V_{ji}(n)$	=	value of weight V_{ji} at n^{th} iteration
$V_{ji}(n+1)$	=	updated value of weight V_{ji} at n^{th} iteration
O_j	=	output from j^{th} neuron in the output layer
α	=	momentum constant
η	=	learning constant

The value of δ_i for output neuron is given by

$$\delta_i = O_i(1-O_i)(D_i-O_i) \qquad (3.6)$$

where O_i = output from the network

D_i = target value of the output

δ_i = error signal term of the output layer

In the output layer, the target outputs are known, in the hidden layers, target values are not known. Therefore, the back-propagation algorithm uses the sum of the error signals of all the neurons of the succeeding layers to calculate error signal of any neuron 'j ' in the hidden layer.

$$\delta_i = O_i(1-O_i)\sum_p \delta_p W_{qp} \tag{3.7}$$

where p runs over all the neurons in the subsequent layers and δ_p is the error signal term corresponding to subsequent layers of p . The value of δ_i is then substituted in the equation 3.5. This procedure is repeated up to the selected accuracy is achieved.

3.5 Activation Function

The output from a neuron is calculated through the use of an activation function. The activation function can be sigmoid, hyperbolic tangent, or sinusoidal. Usually, the sigmoid function is used. The basic characteristics of the sigmoid function are that it is continuous, differentiable and is monotonically increasing. The sigmoid function is shown Figure 3.4.

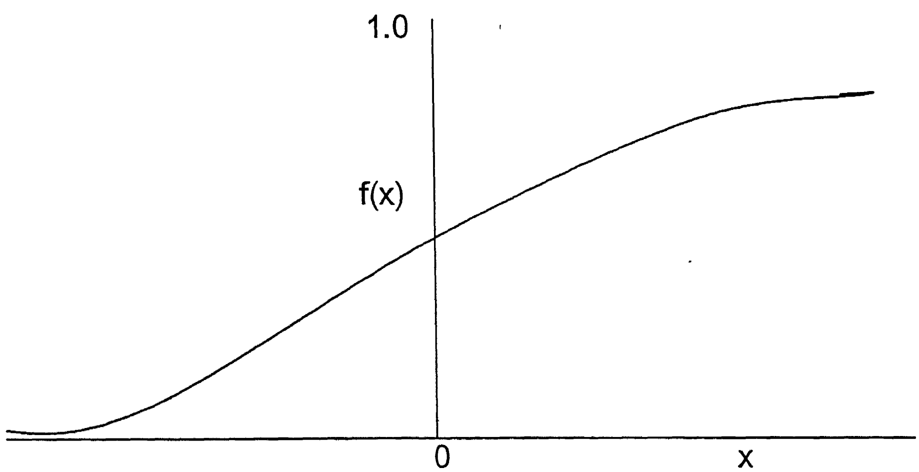


Figure 3.4: Sigmoid Function

The sigmoid function can be represented by the following equation.

$$f(x) = \frac{1}{1+\exp(-ax)} \tag{3.8}$$

wher a= slope parameter.

The output from sigmoid function is always bounded between 0 and 1 and input to the function can vary between $-\infty$ to $+\infty$.

3.6 Initialization of weights

In an ANN, the weights are normally initialized to small random values. The initialization strongly affects the ultimate solution. The motivation for starting from small weights is that large weights tend to prematurely saturate weights in a network and render them insensitive to the learning process. To avoid using the same weights in a network, the randomness is introduced to break the symmetry of weights. However with a random selection of weights we may end up in a local minimum of the error function E, and we may then have to repeat the learning process with other random weights in order to determine whether the final solution is a local minimum or not. In general, all weights be initialized in the ranges ±0.3, ±0.5 or ±0.7 depending up on the particular application. The choice of initial weights is, however only one of several factors affecting the training of the network towards on acceptable error minimum.

3.7 Learning Constant

The rate of convergence in a back-propagation ANN is directly related to the learning constant (η). Selection of a value for the η, has a significant effect on the network performance. Usually, η must be a small number, on the order of 10^{-3} to 10 to ensure that the network will settle to a solution. A small value of η means that the network will have to make a large number of iterations. It is often possible to increase the size of η as learning proceeds. Increasing η as the

network error decreases will often help to speed convergence by increasing the step size as error reached a minimum.

3.8 Momentum Constant

In order to achieve faster convergence and achieve increased stability, a momentum constant is often used, which smoothes out the error correction over time. The moment term determines the effect of previous weight change on the present change in the weight space. Adding a momentum term sometimes results in much faster training. Momentum term is analogous to the moving average process term in the time-series models. Generally, the values of α is chosen between 0 to 1. The momentum constant can speed up training in very flat regions of the error surface and help prevent oscillations in the weights.

3.9 Applications of ANNs in Engineering

ANNs have been used extensively in engineering in recent year. Kirkegaard and Rytter (1993) used ANN for damage detection and location in steel member. Arslan and Ince (1994) applied ANN technique for the design of edge supported reinforced concrete slabs. Barai and Pandey (1995) used ANN for vibration signature analysis. ANNs have been used as computational tools in various areas of structural mechanics (Topping and Bahreininejad 1997). Mingolla, Ross and Grassberg (1999) used ANN approach for enhancing boundaries and surfaces in synthetic aperture radar images. Some other examples include environmental applications for ANNs (Schmuller 1990), optimization of pumping costs (Garret *et al.* 1993), combined fuzzy logic and neural networks for reservoir management (Kojiri *et al.* 1994), forecast water availability using global an solar indices (Zang and Trimble 1995), estimate the snow-water equivalent from the spatial sensor microwave (Sun *et al.* 1995).

Chapter 4

Model Development

4.1 Introduction

Two types of model structures have been developed in this study. The first type of models are time-series models of autoregressive (AR) type and the second type of models are ANN models. The data used in this study consist of monthly runoff at Colorado River at Lees Ferry, U.S.A., for a period of 62 years. First 57 years of data were used for calibration/training and the remaining 5 years of data were used for testing the performance of all the models developed in this study. The performance of all the models was quantified using certain standard statistical parameters. The standard statistical parameters are discussed in next chapter.

4.2 Autoregressive (AR) Models

Autoregressive models may have constant parameters, parameters varying with time or a combination of both. The general steps involved in developing the AR models are explained in following sections.

4.2.1 Modeling for Long-Term Trend

The monthly runoff data can be represented by $X(i,t)$ series, where i varies from 1 to n years and t varies from 1 to 12 months. The first step in the time-series modeling is to investigate for any long-term trends. This can be done by the calculation of annual mean flows for the selected data set. Then an appropriate

function either linear or non linear can be fitted to find the long-term trend of the X(i,t) series. With this function, the long-term component L(i) can be determined, where i represents the i^{th} year in the data set. The detrended series $X_1(i,t)$ can then be found by removing the long-term component from original data series.

$$X_1(i,t) = X(i,t) - L(i) \tag{4.1}$$

4.2.2 Modeling for Seasonality

Once the long-term trends are removed, the next step in time-series modeling is to investigate for any seasonality effects. In order to remove seasonality from a time-series, either arithmetic or fourier mean approach can be used. In the present study, seasonality effects in time-series were removed using fourier mean approach. The fourier mean approach can be represented by the following equations:

$$S(t) = \sum_{k=1}^{K/2} \left[a_k \cos\frac{2\Pi kt}{K} + b_k \sin\frac{2\Pi kt}{K}\right] \tag{4.2}$$

$$a_k = \frac{2}{K}\sum_{t=1}^{K}\frac{1}{n}\sum_{t=1}^{n} X_1(t,i)\cos\frac{2\Pi kt}{K} \quad \text{for } k=1 \text{ to } \frac{K}{2}-1 \tag{4.3}$$

$$= \frac{1}{2} \quad \text{for } k=\frac{K}{2} \tag{4.4}$$

$$b_k = \frac{2}{K}\sum_{t=1}^{K}\frac{1}{n}\sum_{t=1}^{n} X_1(t,i)\cos\frac{2\Pi kt}{K} \quad \text{for } k=1 \text{ to } \frac{K}{2}-1 \tag{4.5}$$

$$= 0 \quad \text{for } k=\frac{K}{2} \tag{4.6}$$

Where a_k , b_k = Fourier coefficients to be determined.

t , k = indices representing periodicity in data (k=1 to K, K=12 in present case)

i = an index representing number of years of the record

(i = 1 to N , N = 57 years in present case)

After developing seasonality component, it can be removed from the detrended series to get detrended deseasonalized series $X_2(i,t)$.

$$X_2(i,t) = X_1(i,t) - S(t) \tag{4.7}$$

4.2.3 Modeling for Auto Correlation Structure

The next step in time-series modeling is to investigate for auto correlation structure for the detrended deseasonalized time-series. The detrended deseasonalized time-series is usually normalized before investigating for the auto correlation function so that it has a mean 0 and standard deviation 1.0. Let it be $X_3(i,t)$ series. Then the resulting $X_3(i,t)$ series can be transformed in to single dimension time-series. Then the series can be investigated for auto correlation function using the following equation.

$$\rho(v) = \frac{\frac{1}{n-v}\sum_{t=1}^{n-v}(X_3(t)-\bar{x})(X_3(t+v)-\bar{x})}{\frac{1}{n}\sum_{t=1}^{n-v}(X_3(t)-\bar{x})^2} \tag{4.8}$$

where v represents the lag

n total number of data sets

$\bar{x}$ mean of the series $X_3(t)$ series

A correlogram can be plotted with autocorrelation coefficients against lag. The auto correlation coefficient is just like a correlation coefficient, therefore has to lie between –1 to 1. If lag is 0 the correlation coefficient is 1. The auto correlation function is used to determine the linear dependence existing in a time-series.

This dependency can be achieved by varying lag from 1 to p. The auto regressive model of order p for $X_3(i,t)$ series is defined as:

$$X_4(t) = \varphi_{p,1} X_3(t-1) + \varphi_{p,2} X_3(t-2) + ...+ \varphi_{p,p} X_3(t-p) + R(t) \qquad (4.9)$$

Where φ is the AR parameter and R(t) is the independent random variable.

The AR parameters can be obtained from Yule Walker equations. The matrix form of Yule Walker equations is given below.

$$\begin{bmatrix} \rho 1 \\ \rho 2 \\ \rho 3 \\ \rho 4 \end{bmatrix} = \begin{bmatrix} 1\rho 1\rho 2\rho 3 \\ \rho 1 1\rho 1\rho 2 \\ \rho 2\rho 1 1\rho 1 \\ \rho 3\rho 2\rho 1 1 \end{bmatrix} \begin{bmatrix} \Phi p,1 \\ \Phi p,2 \\ \Phi p,3 \\ \Phi p,4 \end{bmatrix} \qquad (4.10)$$

Once the auto correlation coefficients are determined, the model is validated and then used for forecasting. This can be done using the following equation:

$$Y_1 (i) = SD (X_4(t)) + mean + S(t) + A(i) \qquad (4.11)$$

Where $Y_1(i)$ = Forecasted value of monthly flow

S(t) = Seasonal component

A(i) = Long term component.

Once the AR parameters were determined using the calibration data set, the model structure were used to compute various standard statistical parameters using the both calibration and testing data sets, in order to evaluate performance of all the models.

4.3 Development of the ANN Model

The general steps involved in developing the ANN model are explained by the following steps. An optimal data set should be representative of the probable occurrence of an input vector and should facilitate the mapping of the underlying non-linear process. Inclusion of unnecessary patterns could slow down the network training. This makes it useful to analyze and preprocess the data before it is used for an ANN application. The data needs to be encoded, normalized before being applied to an ANN.

The important step involves the determination of the ANN architecture and selection of training algorithm. An optimal architecture may be considered the one yielding the best performance in terms of error minimization, while retaining a simple and compact structure. The numbers of input and output nodes are problem dependent. The flexibility lies in selecting the number of hidden layers and in assigning the number of nodes to each of these layers. A trial and error procedure is generally applied to decide on the optimal architecture.

The next step is to train the optimal ANN architecture. The purpose of training is to determine the set of connection weights and thresholds that cause the ANN to estimate outputs that are sufficiently close to target values. The dataset reserved for training is used to achieve this goal. This function of the complete data to be employed for training should contain sufficient patterns so that the network can mimic the underlying relationship between input and output variables adequately. The next step is the performance of a trained ANN can be fairly evaluated by subjecting it to new patterns that it has not been during training. The performance of the network can be determined by comparing forecasted and desired values.

A computer program in C has been developed to simulate back-propagation ANN for runoff modeling in this study. The flow chart for the computer program for simulating a back-propagation ANN is shown in Figure 4.1.

4.4 ANN Models for Runoff Process

Once the computer code is developed for back-propagation ANN, it can simulate any type of ANN architecture. Two types of neural network architectures were developed in this study. First type of architecture called single hidden-layer ANN consists of one input layer, one hidden layer and one output layer. The general form of this architecture is n_1-n_2-1. The developed computer program was used to investigate various types of single hidden-layer ANNs. The various types of single hidden-layer ANNs can be obtained by varying the neurons from 1 to n_1 in the input layer, and to investigate the best ANN model, hidden layer neurons can be varied from 1 to n_2. The number of hidden layer neurons required is much more difficult to determinine, since no general methodology is available for its determination. The number of neurons in the hidden layer of the network was finalized using trial and error procedure. The second model structure was a more complex multi hidden-layer ANN.

The ANN models were developed using three different categories of data sets. These are: a) original monthly runoff data, say category 1 b) detrended monthly runoff data, say category 2 and c) detrended deseasonalized monthly runoff data, say category 3. This was done in an attempt to achieve better performance in modeling the runoff process, by fitting the long-term trend and seasonality trends from the original time-series before presenting the data to the ANN.

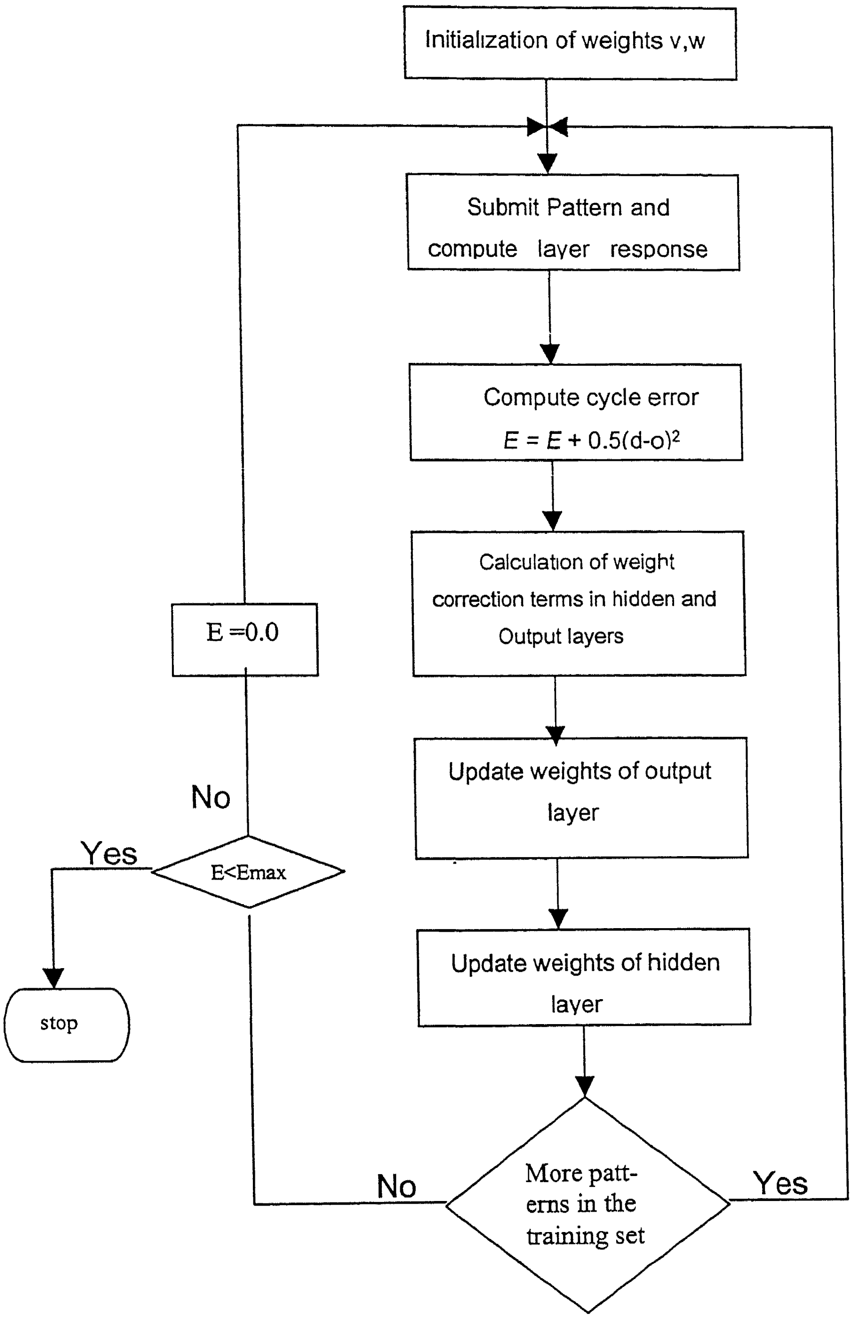


Figure 4.1: Flow Chart of Developed Computer Program

4.5 Single Hidden-Layer ANN Model

A simple model 1-n-1 was selected first to model monthly runoff. In this model, one input neuron in the input layer represents the previous monthly flow to predict the current monthly runoff as an output neuron in the output layer. Various ANN configurations were trained and tested using this model by varying number of neurons in the hidden layer.

In the next model of ANN architecture, the runoff at time t depends on runoff at time steps t-1 and t-2, this leads to an ANN model 2-n-1. With this model, various configurations were trained and tested by varying hidden layer neurons. In order to obtain improved performance this procedure was extended to the runoff at 12 time steps in the past. Various configurations were trained and tested by varying hidden layer neurons and input layer neurons. The best selected models in the single hidden-layer ANNs 1-7-1 and 9-15-1 of category 3 were extended to multi hidden-layer ANNs to achieve better performance. The results for all the ANNs are provided in the appendix and results for best ANNs in each category are discussed in next chapter.

4.6 Multi Hidden-Layer ANN Model

The models 9-15-1 and 1-7-1 have been used for developing multi hidden-layer ANN models based upon their better performance in terms of standard statistical parameters. The structure of the input layer for multi hidden-layer ANN model was same as that of the single hidden-layer ANN model. The general structure of multi hidden-layer ANN model can be represented by $9\text{-}n_1\text{-}n_2\text{-}1$ or $1\text{-}n_1\text{-}n_2\text{-}$

1. Where n1 and n2 are number of neurons in the first and second hidden layers respectively. Various ANN networks were developed to simulate runoff process. The results from these networks in terms of some statistical parameters are presented in the next chapter. Out of all the networks investigated in this study, the model 9-2-12-1 gave the best results. The performance of the 9-2-12-1 model in terms of some statistical parameters, for both training and testing, are presented in the next chapter. The best ANN network obtained by all the trial and error procedures for the simulation of runoff process is given below.

The error in training of a back-propagation ANN was reduced considerably during the initial stages and sets compressed slowly. Once the ANN has been trained, it is ready for prediction. Using the same data set used for training one can check the performance of the trained ANN. We can evaluate the performance of trained ANN in recognizing the patterns that it has not seen before. The results in terms of some statistical parameters both for training and testing sets are presented in the next chapter.

Chapter 5

Results and Discussions

5.1 General

Two types of models have been developed for monthly runoff forecasting process. The first type of models are autoregressive models and second type of models are ANN models. The data collected from Colorado River at Lees Ferry, U.S.A., for the period of 1911-72 were employed in this study. The data from 1911-60 was used for the calibration purposes, while remaining data was used for the testing purposes. The performance of all the models was measured using certain standard statistical parameters, which are discussed next.

5.2 Statistical Parameters

Three types of statistical parameters were used for quantifying the performance of each of the model. These parameters play a dominant role in selecting the best model among all models. These parameters are discussed below.

5.2.1 Average Absolute Relative Error (AARE)

AARE is the average of the absolute values of the relative error in forecasting a number of data points. To find the AARE, we need to first find relative error in forecasting a data point. Relative error is a measure of the error in forecasting a particular variable relative to its exact value. Mathematically, it can be represented by the following equation.

$$RE(t) = \frac{RO(t) - RF(t)}{RO(t)} \; X \; 100\% \tag{5.1}$$

Where $RE(t)$ = Relative error in forecasting

$RO(t)$ = Observed runoff at time t

$RF(t)$ = Forecasted runoff at time t

The relative error *RE(t)* can be either positive or negative. Using relative error *RE(t)*, *AARE* can be evaluated as follows:

$$AARE = \frac{1}{N}\sum_{t=1}^{N} |\mathrm{RE(t)}| \tag{5.2}$$

Where *AARE* = Average Absolute Relative Error

N = Total number of data points forecasted.

It is obvious, lower AARE value represents good model performance, and vice-versa.

5.2.2 Threshold Statistics

It is another important parameter to quantify the performance of a model. It measures the model performance at certain level of absolute relative error say *p* . The threshold statistics can be defined as the percentage of data points predicted for which the absolute relative error is less than a certain level of relative error (say *p%).*

Mathematically Threshold statistic can be represented by

$$TS_p = \frac{n}{N} \; X \; 100\% \tag{5.3}$$

Where n = Number of data points whose absolute relative error is less than p

N = Total number of data points

It is obvious that higher the threshold statistics value better is the model performance and vice versa.

5.2.3 Correlation Coefficient (R^2)

The correlation coefficient measures the correlation between forecasted and observed value of the variable being modeled. Correlation coefficient can be used as a measure of the performance of the model. Higher values indicated good model performance and vice-versa. Mathematically, it can be expressed using following equation:

$$R^2 = \frac{\Sigma X1X2}{\sqrt{\Sigma X1X1}\sqrt{\Sigma X2X2}} \tag{5.4}$$

Where $X1 = X1 - \bar{x}$ and $X2 = X2 - \bar{x}$

X1 is deviation of observed value from its mean and X2 is the deviation of forecasted value from its mean.

5.3 Discussion of Results

The results obtained in terms of various standard statistical parameters are presented in tables 5.1 to 5.10. The statistical parameters were calculated during both calibration data set and testing data set, in order to evaluate the performance of various models during calibration and testing, respectively. The discussion of results is accordingly divided into two parts i.e., results during calibration and results during validation.

5.3.1 Results during Calibration

All the models were developed using three different categories of data, as mentioned earlier. Accordingly, the discussion of results has been further divided into following three parts for both calibration and testing data sets.

5.3.1.1 Results for Data in Category 1

Taking into consideration the calibration set models in category 1, it is found that the least AARE of 28.72% was obtained from 9-2-12-1 ANN model whereas AR(1) model was showing largest AARE of 94.67%. An AARE of 34.51% was observed from 9-15-1 ANN model and AR(4) model captured AARE of 88.52%. Based on the results in terms of AARE during calibration in category 1 it can be concluded that the 9-2-12-1 ANN model learnt better than other models.

The best correlation was shown by the 9-2-12-1 ANN model which with a correlation coefficient value of 0.77 followed by 9-15-1 ANN model with a value of 0.70, whereas AR(1) model was found to be giving the correlation coefficient value of 0.48. so, based on the results in terms of correlation coefficient during

calibration in category 1, it can be concluded that the 9-2-12-1 ANN model performed well.

In terms of threshold statistics, the TS-1 for 9-2-12-1 ANN model was found to be 4.80% whereas 9-15-1 and 1-7-1 ANN models had 2.01% and 1.12% respectively. There was no observation having relative error in forecasting less than 1% from AR(1) and AR(2) models. TS-25 value obtained through 9-2-12-1 ANN model was found to be 25.82% followed by 9-15-1 ANN model with a value of 19.02. Further, approximately 65% forecasted values were having relative error less than 75% for 9-2-12-1 ANN model, whereas AR(1) model having 49.52% forecasted values less than 75%. All the threshold statistics were found to be best in case of 9-2-12-1 ANN model, whereas AR(1) model was showing worst threshold statistics. Hence it can be concluded that 9-2-12-1 ANN model had learnt well in terms of various statistical parameters.

5.3.1.2 Results for Data in Category 2

An AARE of 9.452% was obtained from 9-2-12-1 ANN model whereas AR(1) model was showing largest AARE of 72.76%. 9-15-1 ANN model captured an AARE of 14.14% followed by 1-7-1 ANN model with a value of 18.97%. Based on results in terms of AARE, it can be concluded that the 9-2-12-1 ANN model learnt better than other models.

The correlation coefficient of 0.89 was obtained with the 9-2-12-1 model followed by 9-15-1 ANN model with a value of 0.82. AR(4) gave least correlation coefficient of 0.51. So based on results in term of correlation coefficient, it can be concluded that the 9-2-12-1 ANN model learnt better than other models.

The TS-5 for 9-2-12-1 ANN model was found to be 9.530% and the same was found to be 8.82% from the 9-15-1 ANN model. About 28% forecasted values were having relative error less than 25% for 9-2-12-1 ANN model whereas AR(4) model was having 12.152% forecasted values less than 25%. TS-75 value was achieved 70.42% for 9-2-12-1 ANN model followed by 1-7-1 ANN model with a value of 58.42%. Overall the threshold statistics were found to be the best in the case of 9-2-12-1 ANN model. Hence we may conclude that 9-2-12-1 ANN model performed the best in terms of all the statistical parameters.

5.3.1.3 Results for Data in Category 3

AARE of 1.017% was obtained from 9-2-12-1 ANN model whereas AR(1) model captured largest AARE of 36.74%. The 9– 15-1 ANN model achieved an AARE of 4.14% and 35.89% of AARE was observed in the case of AR(4) model. Based on the results in terms of AARE it, can be concluded that the 9-2-12-1 ANN model learnt better than other models.

The best correlation was achieved by the 9-2-12-1 ANN model with a value of 0.989 followed by 9-15-1 ANN model with a value of 0.912, whereas AR(1) and AR(4) models captured 0.72 and 0.73 respectively. Based on the result in terms of correlation coefficient it can be concluded that the 9-2-12-1 ANN model learnt better than other models.

In 13.3% of forecasted monthly runoff values the relative error was less than 0.5% from AR(4) and AR(3) models whereas 9-2-12-1 ANN model was having 7.22% forecasted values less than 0.5%. The TS-5 for 9-2-12-1 ANN model was found to be 45.37% whereas 9-15-1 and 1-7-1 ANN models had 37.82% and 30.87%. AR(4) model followed the 1-7-1 ANN model with a value of

22.22% for TS-5. The largest value of TS-100 of 98.85% was achieved from 9-2-12-1 ANN model whereas AR(1) model was showing 95.62%. In terms of threshold statistics 9-2-12-1 ANN model was found to be the best model. Hence we may conclude that 9-2-12-1 ANN model had learnt well as judged by the statistical performance.

5.3.2 .Results During Validation

The discussions were also made separately for data sets category 1, category 2 and category 3 during validation.

5.3.2.1 Results for Data in Category 1

An AARE of 38.812% was observed from 9-2-12-1 ANN model whereas AR(1) model was showing largest AARE of 112.34%. 9-15-1 ANN model followed the 9-2-12-1 ANN model with an AARE of 41.52%. The AR(4) model achieved approximately 100% of AARE whereas 1-7-1 ANN model showing AARE of 61.98%. Based on the results in terms of AARE it can be concluded that the 9-2-12-1 ANN model learnt better than other models.

The largest correlation coefficient value was obtained by the 9-2-12-1 ANN model with a value of 0.68 followed by 9-15-1 ANN model with a value of 0.62, whereas the least correlation coefficient 0.302 was achieved from AR(1) model. Based on the results in terms of correlation coefficient it can be concluded that the 9-2-12-1 ANN model learnt better than other model.

There was no observation having relative error in forecasting less than 0.5% in all the models investigated in this category. TS-1 for 9-2-12-1 ANN model was found to be 1.52% whereas there was no observation having relative error less than 1% in all the other models. The 9-2-12-1 ANN model achieved 42.18% for TS-50 whereas AR(4) model was showing only 22.12%. In terms of threshold statistics 9-2-12-1 ANN model was found to be the best model. Hence we may conclude that 9-2-12-1 ANN model had performed well as judged by the statistical performance during testing in category 1.

5.3.2.2 Results for Data in Category 2

An AARE of 18.26% was observed from 9-2-12-1 ANN model whereas 28.72% of the AARE was obtained from 9-15-1 ANN model. 1-7-1 ANN model achieved an AARE of 41.86% followed by AR(4) model with a value of 87.52%. Based on the results in terms of AARE, it can be concluded that the 9-2-12-1 model learnt better than other models.

The correlation coefficient of 0.77 was obtained with the same model followed by 9-15-1 ANN model with a value of 0.70. The AR(4) model captured correlation coefficient of 0.50. Based on the results in terms of correlation coefficient it can be concluded that the 9-2-12-1 ANN model learnt better than other models.

In terms of threshold statistics, TS-1 for 9-2-12-1 ANN model was found to be 4.80% whereas 9-15-1 and 1-7-1 ANN models had 3.72% and 2.620%. No observation was achieved relative error in forecasting less than 1% in AR(1) and AR(2) models. TS-50 value obtained through 9-2-12-1 ANN model was found to be 42.18% followed by 9-15-1 ANN model with a value of 40.72%. 65.42% forecasted values were having relative error less than 75% for 9-2-2-1 ANN model whereas AR(2) model having 45.12% forecasted values less than

75%. Overall the threshold statistics were found to be the best in the case of 9-2-12-1 ANN model. Hence we may conclude that 9-2-12-1 ANN model performed the best in terms of all the statistical parameters.

5.3.2.3 Results for Data in category 3

The least AARE of 7.39% was achieved from 9-2-12-1 ANN model followed by 9-15-1 ANN model with a value of 15.45% whereas AR(4) model obtained an AARE of 57.16%. An AARE of 33.26% was obtained from the 1-7-1 ANN model. Based on the results in terms of AARE it can be concluded that the 9-2-12-1 ANN model learnt better than other models.

The best correlation coefficient achieved from the same model with a value of 0.82 followed by 9-15-1 ANN model with a value of 0.80. The correlation coefficient of 0.67 was observed from AR(4) model whereas AR(1) model gave 0.60. Based on the result in terms of correlation coefficient it can be concluded that the 9-2-12-1 ANN model learnt better than other models.

In terms of threshold statistics, in 44.826% of forecasted monthly runoff values the relative error was less than 5% from 9-2-12-1 ANN model whereas AR(4) model having 21.961% of forecasted monthly runoff values less than 5%. The TS-50 for 9-2-12-1 ANN model was found to be 87.182% whereas 9-15-1 ANN and AR(4) models had 77.729% and 42.871%. AR(3) model obtained 52.575% of TS-75 whereas AR(1) model achieved 51.18% of forecasted monthly runoff values having the relative error was less than 75%.

All the threshold statistics were found to be best in case of 9-2-12-1 ANN model. Hence it can be concluded that 9-2-12-1 ANN model had learnt well in terms of various statistical parameters.

During validation of all the models investigated the 9-2-12-1 ANN model performed the best in terms all the statistical parameters.

Observed and forecasted values of various models for calibration and validation for category 3 are shown figures: 5.1 to 5.6. The 9-2-12-1 ANN model match the observed values most closely in calibration, whereas AR(1) model was showing significant deviations from the observed values. The AR(4) and AR(3) models were having less deviations compare with AR(1) and AR(2) models in calibration as well as in validation.

5.4 Voting Analysis

A voting analysis was carried out to select the best model among all the models investigated in the study. In this voting analysis a model which performed the best in terms a of particular statistics, receives one vote. Total number of votes available is 22. The ANN model 9-2-12-1 received 19 out of 22 votes i.e., 85% of the total votes in category 1. The same model received 18 out of 22 votes i.e., 80% of total votes in category 2. Out of 22 votes 16 votes received by the 9-2-12-1 ANN model in category 3. Over all the 9-2-12-1 ANN model is deemed to be the best model developed in this study.

5.5 Comparison of ANN Models with AR Models

In order to evaluate the suitability of a technique in monthly runoff modeling was comparison was made on averaged values of statistics from all the models investigated during training and testing. The results of this comparison are presented in table 5.10. An AARE of 60.36% was observed from AR models during calibration whereas multi hidden-layer ANN captured an AARE of 12.862% followed by single hidden-layer with a value of 30.85%. During validation the AARE of 83.12% was achieved from AR models whereas multi hidden-layer ANN obtained 21.62% followed by single hidden-layer ANN with a value of 42.16%. Hence it can be concluded that multi hidden-layer ANN model performed the best in terms of AARE both during calibration and validation.

The best correlation coefficient was obtained from multi hidden-layer ANN model with a value of 0.88 during calibration whereas least correlation coefficient of 0.63 was observed from AR model. During validation also multi hidden-layer ANN model performed best in terms of correlation coefficient. Hence it can be concluded that multi hidden-layer ANN model performed best in terms of correlation coefficient both during calibration and validation.

In terms of threshold statistics, the TS-5 for multi hidden-layer ANN model was found to be 20.15% during calibration whereas AR and single hidden-layer ANN models had TS-5 values 8.26% and 12.34%. During validation also, multi hidden-layer ANN model was showing the best forecasted monthly runoff values of 18.26% the relative error less than 5%. In 60.16% of forecasted monthly runoff values the relative error was less than 25% from multi hidden-layer ANN model during calibration, whereas AR and single hidden-layer ANN models were showing 52.18% and 27.56% forecasted values less with absolute relative error

then 25%. During validation also multi hidden-layer ANN model was showing the best forecasted monthly runoff values of 42.98% having the relative is less than 25%. In terms of threshold statistics multi hidden-layer ANN model was found to be the best model. Hence it can be concluded that multi hidden-layer ANN model learnt well as judged by the statistical performance.

Table 5.1 Statistical Performance of Models for Category 1

MODEL	AARE	CORRELATION COEFICIENT (R^2)
TRAINING		
AR(1)	94.67	0 48
AR(2)	92.78	0.48
AR(3)	90.42	0.50
AR(4)	88.52	0.51
1 – 7 – 1	41.31	0.61
2 – 8 – 1	44.51	0.62
3 – 9 – 1	41.72	0.66
4 – 9 – 1	44.01	0.68
9 – 15 – 1	34.51	0.70
9 – 2 – 12 - 1	28.72	0.77
TESTING		
AR(1)	112.34	0.30
AR(2)	110.42	0.31
AR(3)	104.72	0.38
AR(4)	100.76	0.40
1- 7 – 1	61.98	0.51
2 – 8 - 1	65.52	0.52
3 – 9 – 1	60.42	0.55
4 – 9 – 1	55.52	0.59
9 – 15 – 1	41.52	0.62
9 – 2 - 12 - 1	38.81	0.68

Table 5.2 Statistical Performance of Models for Category 2

MODEL	AARE	CORR COEF (R^2)
TRAINING		
AR(1)	72.76	0.51
AR(2)	71.67	0.52
AR(3)	70.79	0.57
AR(4)	68.62	0.59
1 – 7 – 1	18.97	0.78
2 – 9 – 1	19.55	0.77
3 – 9 – 1	19.11	0.77
4 – 13 – 1	17.67	0.80
9 – 15 – 1	14.14	0.82
9 – 2 – 12 - 1	9.452	0.89
TESTING		
AR(1)	89.92	0.42
AR(2)	89.13	0.45
AR(3)	88.52	0.48
AR(4)	87.52	0.50
1 - 7 - 1	41.86	0.61
2 – 9 - 1	42.01	0.66
3 – 9 – 1	41.18	0.67
4 – 13 – 1	39.62	0.69
9 – 15 – 1	28.72	0.70
9 – 2 - 12 - 1	18.26	0.77

Table 5.4 Statistical Performance of Models in Training for Category 1

MODEL	TS_0.5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
AR (1)	0.00	0.00	1.02	2.32	4.56	12.52	25.62	49.52	58.65
AR(2)	0.00	0.00	1.60	3.82	6.78	14.26	27.52	50.12	60.12
AR(3)	0.00	1.89	2.89	4.89	8.52	15.52	30.58	52.72	61.625
AR(4)	0.00	1.99	3.12	6.15	10.26	16.26	32.87	53.72	62.76
1 – 7 – 1	0.00	1.12	2.87	4.72	8.817	15.82	30.72	55.76	65.56
2 – 8 – 1	0.00	1.87	3.01	5.72	9.71	16.90	32.81	57.82	66.87
3 - 9 – 1	0.00	1.92	3.01	5.92	10.02	17.03	33.82	58.85	67.98
4 – 9 – 1	0.00	1.98	3.18	6.37	11.62	18.92	34.09	58.97	68.29
9 – 15 – 1	1.24	2.01	3.82	6.27	12.81	19.02	36.72	59.21	69.19
9 – 2 – 12 - 1	3.25	4.80	6.53	10.62	19.82	25.82	42.18	65.49	71.82

Table 5.5 Statistical Performance of Models in Testing for Category 1

MODEL	TS_0.5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
AR (1)	0.00	0.00	0.00	0.00	1.28	2.57	15.82	29.58	47.42
AR(2)	0.00	0.00	0.00	0.00	1.92	4.262	17.57	30.14	50.25
AR(3)	0.00	0.00	0.00	1.89	2.52	5.55	20.72	32.90	55.85
AR(4)	0.00	0.00	0.00	2.15	3.09	6.26	22.19	36.10	55.91
1 – 7 – 1	0.00	0.00	1.92	3.72	6.02	8.61	28.72	45.09	61.97
2 – 9– 1	0.00	0.00	2.07	4.92	6.02	10.02	28.12	47.99	62.01
3 - 9 – 1	0.00	0.00	2.70	5.23	8.01	12.19	33.01	47.86	61.98
4 – 13 1	0.00	0.00	3.18	6.37	10.62	12.9	34 09	48.99	63.22
9 – 15 – 1	0.00	0.00	2.81	6 27	12.81	14.02	36.72	59.28	62.19
9 – 2 – 12 - 1	0.00	1.529	3.53	8.62	19.82	23.82	42.18	65.42	68.82

Table 5.6 Statistical Performance of Models in Training for Category 2

MODEL	TS_0.5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
AR (1)	0.00	0.00	1.62	2.39	6.56	12.15	25.62	50.21	65.71
AR(2)	0.00	0.00	1.70	4.71	7.01	14.26	27.52	52.09	65.83
AR(3)	0.00	1.98	2.89	5.01	8.62	16.91	32.54	54.76	66.83
AR(4)	0.00	1.99	3.12	6.98	10.26	17.92	33.90	55.65	67.63
1 – 7 – 1	0.00	2.20	4.92	7.01	8.817	15.82	30.72	58.46	75.01
2 – 8 – 1	0.00	2.87	5.01	8.01	11.73	18.92	32.81	60.82	76.92
3 - 9 – 1	0.00	2.92	5.61	8.92	14.32	19.06	33.82	61.83	76.98
4 – 9 – 1	0.00	3.08	6.18	9.37	15.62	18.92	34.09	62.92	78.29
9 – 15 – 1	1.64	4.01	8.82	12.27	19.51	25.42	44.72	69.21	80.19
9 – 2 – 12 - 1	3.29	4.80	9.5	14.62	19.82	28.02	46.01	70.42	85 89

Table 5.7 Statistical Performance of Models in Testing for Category 2

MODEL	TS_0.5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
AR (1)	0.00	0.00	1.92	2.38	5.56	12.52	27.62	44.52	56.02
AR(2)	0.00	0.00	1.96	3.82	6.78	13.26	28.52	45.12	56.34
AR(3)	0.00	1 91	2.89	4.89	8.52	16..52	29.58	47.75	56.72
AR(4)	0.00	2.09	3.12	6.15	10.26	16.26	32.81	48.72	56.91
1 – 7 – 1	0.00	2.62	2.87	8.72	10.81	17.12	30.72	55.72	65.72
2 – 8 – 1	0.00	2.87	3.01	9.72	10.91	18.26	32.81	57.82	66.62
3 - 9 – 1	0.00	2.92	3.35	10.92	13.02	17.03	33.82	58.83	67.98
4 – 9 – 1	0.00	3.08	4.18	10.37	14.62	18.92	34.09	58.92	68.29
9 – 15 – 1	1.24	3.72	4.82	11.27	15.81	25.02	40.72	59.28	70.19
9 – 2 – 12 - 1	3.29	4.801	6.53	12.6	19.82	26.82	42.18	65.42	73.75

Table 5.8 Statistical Performance of Models in Training for Category 3

MODEL	TS_0.5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
AR (1)	2.22	4.44	13.14	20.55	30.56	49.52	76.11	89.52	95.62
AR(2)	2.59	4.44	12.22	20.18	31.78	48.26	75.52	89.12	95.94
AR(3)	13.33	15.18	20.74	30.01	37.54	55.52	79.81	91.72	95.62
AR(4)	13.33	15.18	22.22	29 07	36.63	54.26	79.81	92.72	96.71
1 – 7 – 1	7.193	17.70	30.87	62.72	85 81	94.82	97.10	97.72	97.82
2 – 8 – 1	6.29	12.87	31.01	55.72	80.25	89.92	95.81	97.82	97.88
3 - 9 – 1	6.87	13.92	30.01	58.92	81.02	87.03	96.82	97.83	98.18
4 – 9 – 1	8.24	12.89	31.18	60.37	81.62	88.92	96.02	97.92	98.298
9 – 15 – 1	9.45	15.50	37.82	60.27	80.81	92.02	95.12	96.28	97.19
9 – 2 – 12 - 1	7.22	15.55	45.37	66.29	87.12	94.81	98.17	98.42	98.85

Table 5.9 Statistical Performance of Models in Testing for Category 3

MODEL	TS_0.5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
AR (1)	5.88	17.25	28.02	39.0	40.12	42.52	49.35	51.18	58.62
AR(2)	4.30	10.72	15.32	21.87	24.62	34.26	45.68	50.12	60.14
AR(3)	6.54	14.51	24.51	25.0	27.50	35.52	47.58	52.72	61.62
AR(4)	5.49	10.98	21.96	25.21	28.56	36.26	42.87	53.72	62.71
1 – 7 – 1	1.09	8.24	16.42	25.39	46.62	52.82	61.72	80.72	83.42
2 – 8 – 1	2.88	8.12	25.62	32.78	48.98	59.92	64.81	79.01	86.82
3 - 9 – 1	2.71	7.25	28.97	40.73	49.72	55.03	63.83	78.71	89.18
4 – 9 – 1	1.92	7.98	30.72	41.26	50.62	58.92	64.09	80.92	88.29
9 – 15 – 1	2.73	20.62	35.62	44.72	51.81	70.02	77.72	88.28	89.82
9 – 2 – 12 - 1	2.76	12.48	44.82	66.92	79.72	80.82	87.18	92.47	94.82

Table 5.10 Average Statistics of Models

MODEL	AARE	R^2	TS_0. 5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
Training											
Auto Regressive	60.36	0.63	2.87	4.62	8.26	8.26	16.26	27.56	43.82	62.97	78.38
Single Hidden Layer ANN	30.85	0.73	3.31	6 73	12.34	25.62	40.17	52.18	63.02	72.81	86.32
Multi Hidden Layer ANN	12.62	0.88	4.25	8.20	20.15	30.26	44.26	60.16	70.16	80.72	90.02
Testing											
Auto Regressive	83.12	0.50	1.25	4.12	8.52	11.23	16.27	26.73	33.28	44.21	57.28
Single Hidden layer ANN	42.16	0.66	1.00	3.26	11.73	18.34	28.82	39.87	45.83	53.29	68.92
Multi Hidden Layer ANN	21.62	0.78	2.01	6.27	18.26	29.83	38.72	42.98	55.28	64.82	79.87

Table 5.13: Voting Analysis for Category 1

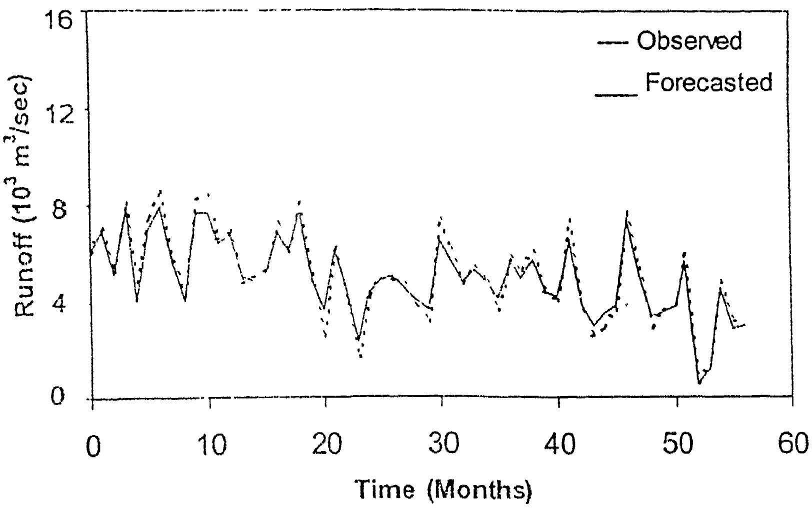
MODEL	VOTES
AR(1)	0
AR(2)	0
AR(3)	0
AR(4)	1
1- 7 - 1	1
2- 8 - 1	0
3- 9- 1	0
4- 9 - 1	0
9- 15 – 1	1
9- 2 -12 – 1	19

Table 5.14: Voting Analysis for Category 2

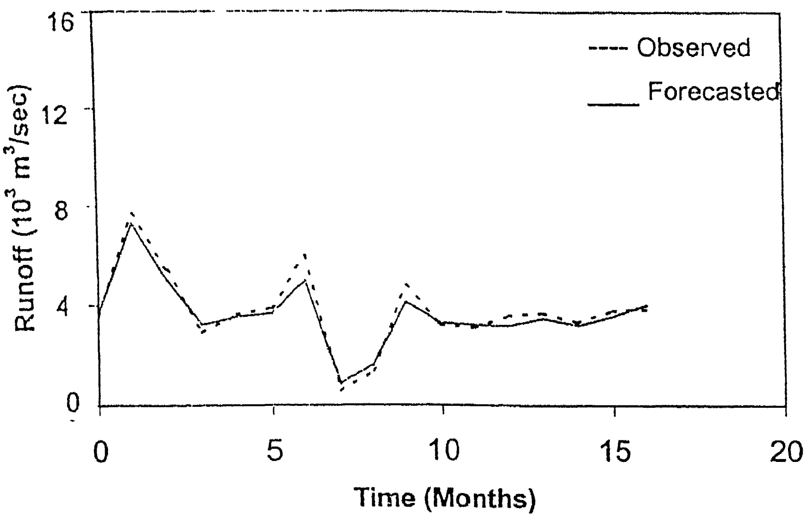
MODEL	VOTES
AR(1)	0
AR(2)	0
AR(3)	1
AR(4)	1
1- 7 - 1	1
2- 9 - 1	0
3- 9- 1	0
4- 13 - 1	0
9- 15 – 1	1
9- 2 -12 – 1	18

Table 5.14: Voting Analysis for Category 3

MODEL	VOTES
AR(1)	0
AR(2)	0
AR(3)	2
AR(4)	1
1- 7 - 1	2
2- 9- 1	0
3- 9- 1	0
4- 12- 1	0
9- 15 – 1	1
9- 2 -12 – 1	16

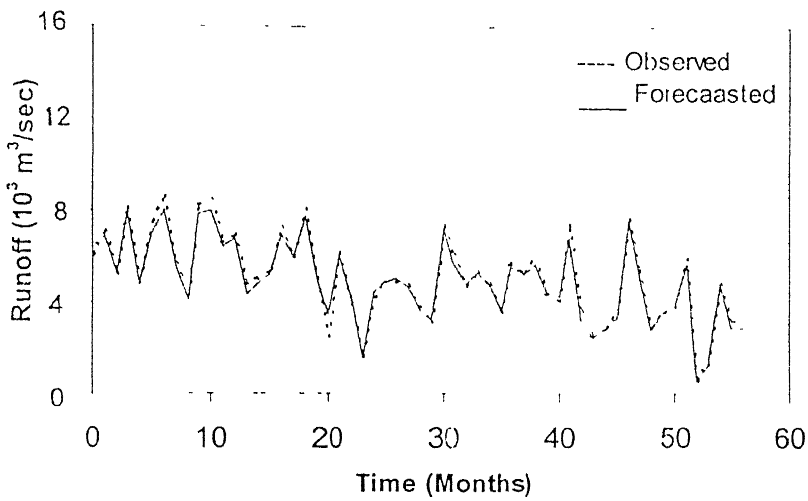


During Training

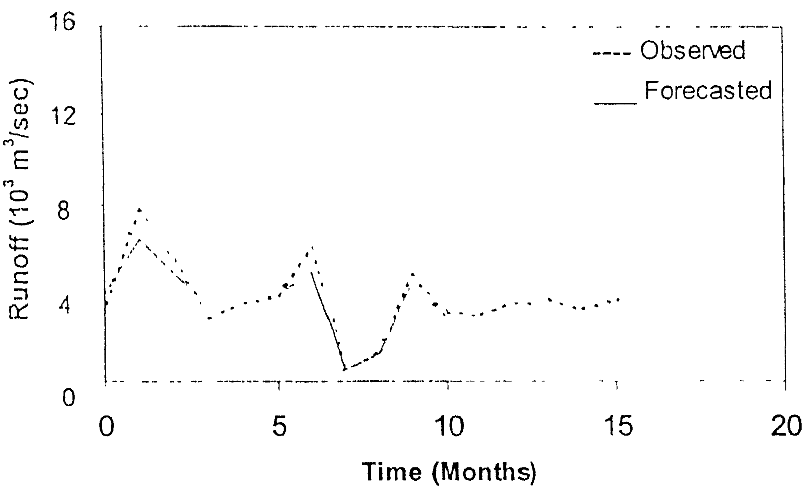


During Testing

Figure 5.5 Observed and Forecasted Runoff from 9-15-1 ANN Model

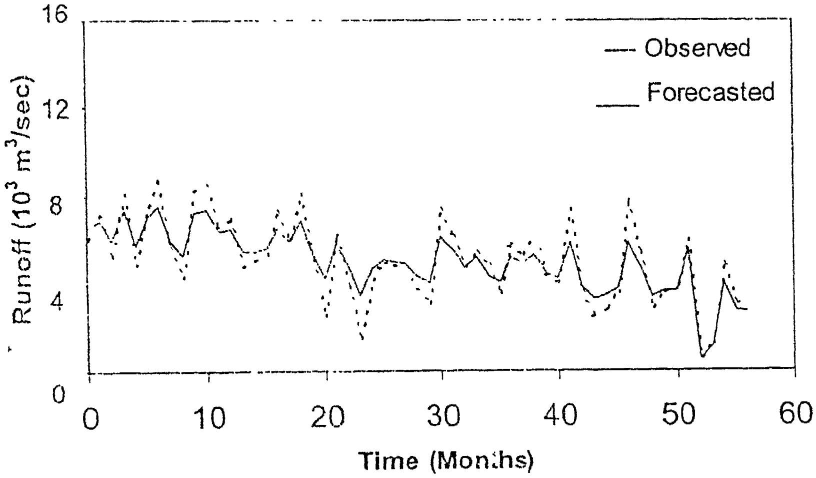


During Training

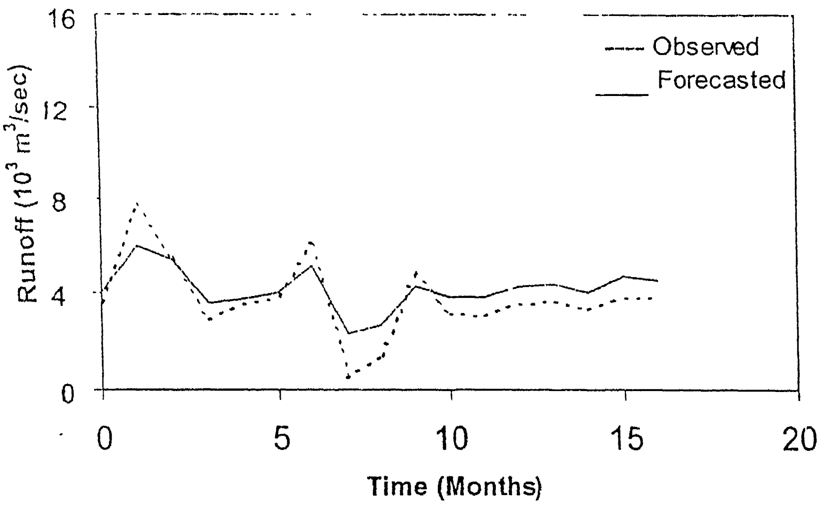


During Testing

Figure 5.6 Observed and Forecasted Runoff from 9-2-12-1 ANN Model

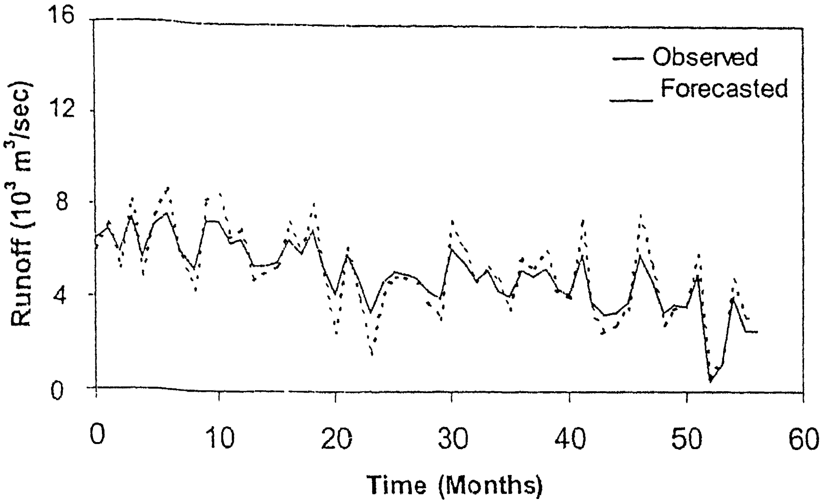


During Training

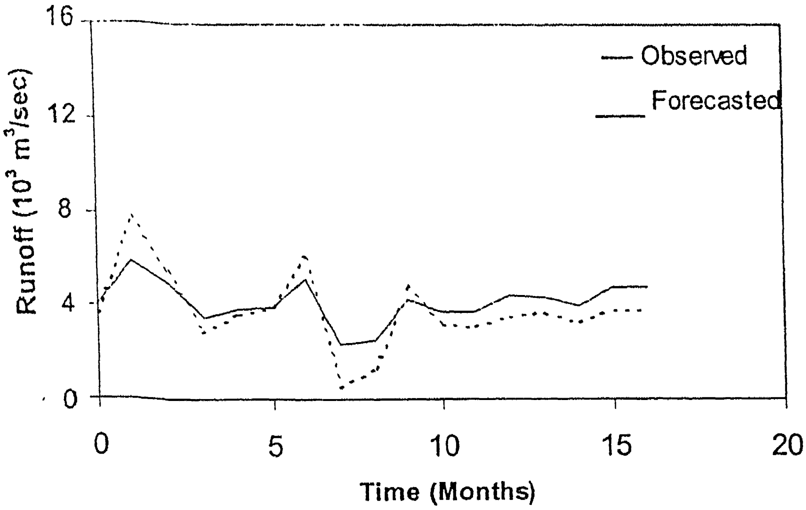


During Testing

Figure 5.4 Observed and Forecasted Runoff from AR(4) Model

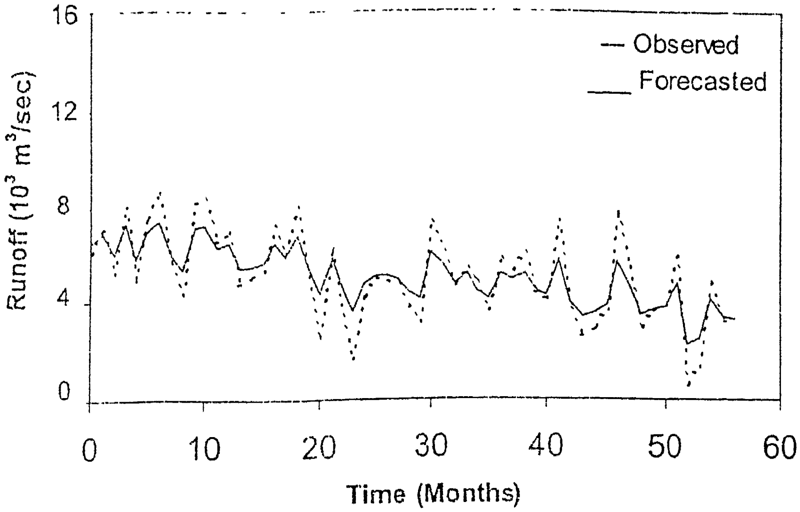


During Training

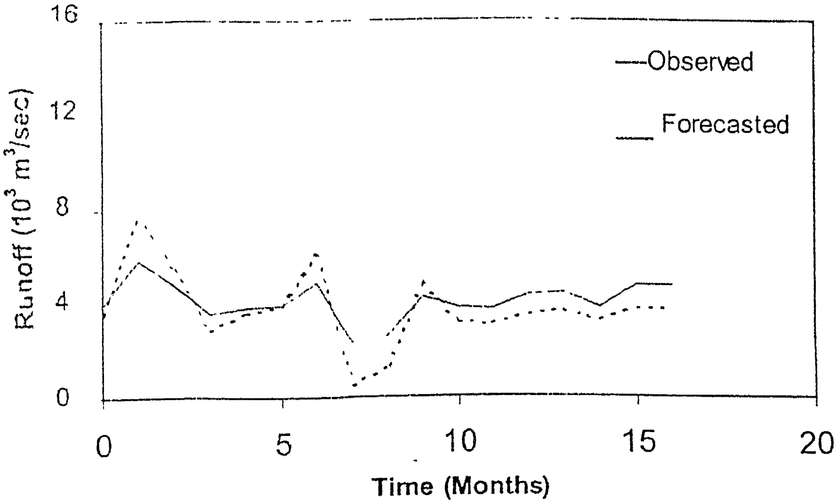


During Testing

Figure 5.3 Observed and Forecasted Runoff from AR(3) Model

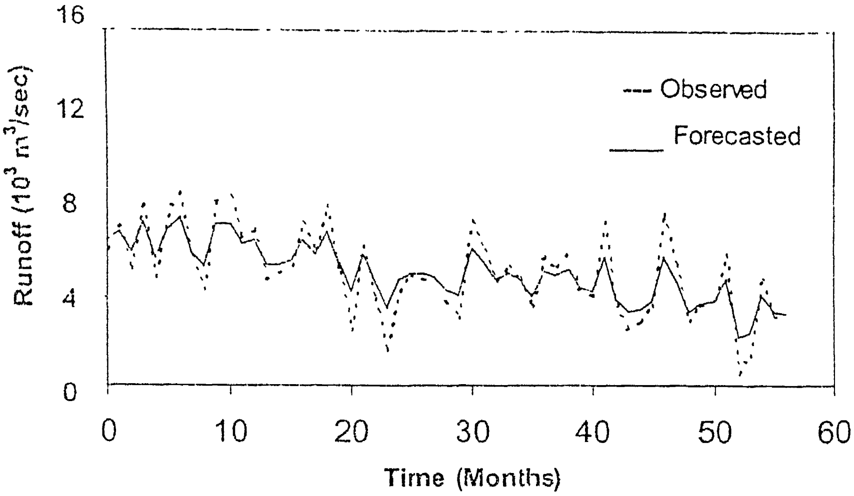


During Training

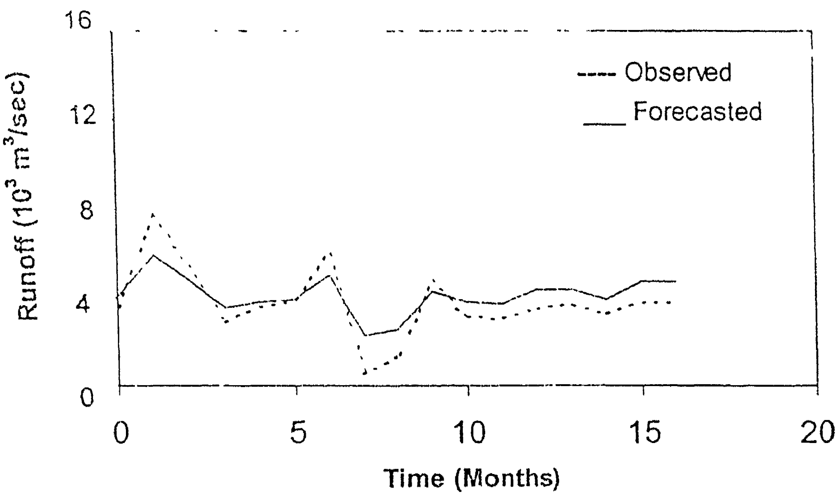


During Testing

Figure 5.2 Observed and Forecasted Runoff from AR(2) Model



During Training



During Testing

Figure 5.1 Observed and Forecasted Runoff from AR(1) Model

Chapter 6

Conclusions

In this study, two types of model structures have been investigated for use in runoff modeling. First type of models are AR models, the second type of models are ANN models. Time-series models such as AR models of order up to 4 were developed. And then ANN technique was applied to different time-series of runoff such as original data, detrended data and detrended deseasonalized data. Different single hidden-layer ANNs were investigated for all the three categories. To achieve that better performance the best single hidden-layer ANNs i.e., 1-7-1 and 9-15-1 were used to develop multi hidden-layer ANNs. All the results are provided in the appendices. The performance of various model structures was evaluated using standard statistical parameters. Based on the results obtained in this study, ANN models have consistently out performed the AR models.

ANN is a relatively new technique that can be used for modeling and forecasting. In the present study, the results achieved are quite encouraging and consistent enough to be used for forecasting purposes. The monthly runoff flows on Colorado River at Lees Ferry, U.S.A., were used in modeling and forecasting in the present study. Obviously, the runoff is dependent on rainfall. In the present study rainfall data is not available for forecasting. It may be possible to achieve more accurate results, when rainfall is available. This is an area that needs further research.

In this study, the back-propagation training algorithm was used for training all the ANN models investigated. The major limitations of back-propagation algorithm are, easily trapped by the local minima, convergence is slow process and quite sensitive to the initial starting point. It may by possible to develop a better ANN model for runoff process with other training algorithms such as, radial basis functions, genetic algorithms, unsupervised algorithm and fuzzy logic *etc.* It is hoped that further research efforts will concentrate in some of these areas.

Bibliography

1) Anderson, H.G., and Burt, T.P. (1985). "Hydrologic Forecasting." John Wiley & Sons Publications.

2) Bender, M. and Simonovic, S. (1992). "Time-series Modeling for long-range stream-flow forecasting." J. of Water Resour. Plng. and Mgmt., ASCE, 120(6), 857-869

3) Black, P.E. (1991). "Watershed Hydrology." Prentice Hall Advanced Reference Series.

4) Bowles, D.S., and Enda O'Connel, P. (1991). "Recent Advances in Modeling of Hydrologic Systems." Kluwer Academics Publications.

5) Crawford, N.H. and Linsely, R.K. (1960). "Digital simulation in Hydrology: Stanford Watershed Model IV." Technical Report 39, Department of Civil Engineering, Stanford University.

6) Dibike, Y.B. and Abbott, B.M (1999). "Application of Artificial Neural Networks to the simulation of a two dimensional flow." J. of Hydraulic Research, IAHR, 37(4), 435-446.

7) Duan, Q., Sorroshian, S. and Gupta, V.K. (1994). "Optimal use of the SCE-UA global optimization method for calibrating watershed models." J. of Hydrology, ASCE, 158, 265-284.

8) Freeman, J.A. (1992). "Neural Networks." Library of Congress Cataloging-in-Publication data.

9) Govindaraju, S.R. (2000). "Artificial Neural Networks in Hydrology." J. of Hydrologic Engrg., ASCE, 5(2), 115-136.

10) Hall, M.J. (1984). "Urban Hydrology." Elsevier Applied Science Publications.

11) Haritham, M., Juan, B., and Pedro, J. (1995). "Streamflow Forecasting for Han River Basin, Korea." J. of Water Resour. Plng. and Mgmt., ASCE, 120(5), 651-673.

12) Hipal, K.w. and Mc Lead. (1994). "Time-series modeling of water resources and environmental systems. " Elsevier science publications.

13) Jain, M.K. and Seth, S.M. (1997). "Rainfall Runoff modeling of a Himalayan catchment using a distributed approach." Institution of Engineers (India) CV, 78, 89-92.

14) Jain, S.K., Das, A., Srivastava, D.K. (1999). "Application of ANN for reservoir inflow prediction and operation." J. of Water Resour. Plng. and Mgmt., ASCE, 120(5), 263-270.

15) Jenkins, G.M. and Box, G.E. (1926). "Time-Series Analysis: Forecasting and Control." Holden-Day, Oakland, California, U.S.A.

16) John, P.G. (1995). "Application of Neural Networks in stratified flow stability analysis." J. of Hydraulic Engrg., ASCE, 121(7), 523-532.

17) Karunanidhi, N., Grenney, W.J and Darrel, W (1994). "Neural Networks for river flow prediction." J. of Computing in Civil Engrg., ASCE, 8(2), 210-221.

18) Macro, J.B., Havboe, R., and Sales, J.D. (1989). "Stochastic Hydrology and its use in Water Resources Systems Simulation and Optimization." Kluwer Academic Publishers.

19) Morshed, J. and Kaluarachichi, J.J. (1998). "Application of Artificial Neural Networks and genetic algorithm inflow and transport simulations." Advances in Water Resources, Elsevier science Ltd., 22(2), 145-158.

20) Ramos, J., Mallants., D. and Feyen, J. (1995). "State space identification of linear deterministic rainfall-runoff models." Water Resources Research, 31(6), 1519-1531.

21) Rodriguez, I and Bras, G.E. (1985). "Random functions and Hydrology." Addition-Wesley, Reading,Massachessets.

22) Salas, J.D. (1993). "Analysis and modeling of hydrologic time series." Chapter 19, Hand book of hydrology."

23) Salas, J.D., Sarada, N and Chung, C.H. (1995). "Stochastic modeling and simulation of the Nile River systems monthly flows". Computing Hydro lab engineering research.

24) Salas, J.D. and Obeysekera J.T.B. (1992). "Conceptual basis of seasonal stream flow time series models". J. of Hydrologic engrg., ASCE, 118(8).

25) Sen, Z. (1976). "Wet and Dry periods of annual flow series". J. of Hydraulic division.

26) Shen, H.W. (1984). "Modeling of Rivers." A Wiley-interscience Publications.

27) Singh, V.P. (1988). "Hydrologic Systems Rainfall-Runoff Modeling Volume-1." Prentice Hall, Englewood.

28) Smith, J. and Eli, R.N. (1995). "Neural Network models of rainfall-runoff process." J. of Water Resour. Plng. and Mgmt., ASCE, 121(6), 499-508.

29) Sunil, K.S. and McKim, R.A. (2000). "Artificial Neural Networks for measuring organizational effectiveness." J. of Computing in Civil Engrg., ASCE, 14(1), 9-14.

30) Todini, E. (1996). "The ARNO rainfall-runoff model." J. of Hydrology., Elsevier Science Publishers, 175, 339-382.

31) Tokar, A.S. and Peggy, A.J. (1999). "Rainfall-Runoff modeling using Artificial Neural Networks." J. of Hydrologic Engrg., ASCE, 4(3), 232-240.

32) Tokar, A.S., Markus, M. (2000). "precipitation-Runoff modeling using Artificial Neural Networks and conceptual models." J. of Hydrologic Engrg., ASCE, 5(2), 156-160.

33) Topping, B.H.V., and Bahreinejad, A. (1997). "Neural Computing for Structural Mechanics." Coburg Publications, Edinburgh.

34) Xu, C.Y. and Vandewiele, G.L. (1995). "Parsimonious monthly rainfall-runoff models for humid basins with different input requirements." Advances in Water Resources, Elsevier Science Limited, 18, 39-48.

35) Wasserman, P.D. (1989). "Neural computing theory and practice." New York: Van Nostrand Reinhold.

36) Zurada, J.M. (1997). "Introduction to Artificial Neural Systems." Jaico Publishing House.

APPENDICES

Results for the Raw (Category 1) Data

Model	AARE	Model	AARE	Model	AARE
1-6-1	45.19	2-7-1	44.88	3-7-1	43.13
1-7-1	41.31	2-8-1	44.51	3-8-1	43.52
1-8-1	42.42	2-9-1	44.54	3-9-1	41.72
1-9-1	41.42	2-10-1	45.16	3-10-1	42.74
1-10-1	41.13	2-11-1	45.96	3-11-1	42.36
1-11-1	42.21	2-12-1	47.56	3-12-1	42.54
1-12-1	43.78	2-13-1	47.92	3-13-1	41.90
1-13-1	43.30	2-14-1	44.55	3-14-1	41.63
1-14-1	44.42	2-15-1	46.66	3-15-1	41.87
1-15-1	44.23	2-16-1	47.96	3-16-1	41.58
1-16-1	44.82	2-17-1	48.55	3-17-1	39.74
1-17-1	45.22	2-18-1	46.16	3-18-1	40.95
1-18-1	46.42	2-19-1	46.86	3-19-1	41.72
4-7-1	44.64	5-8-1	43.34	6-8-1	42.75
4-8-1	44.32	5-9-1	42.74	6-9-1	41.39
4-9-1	44.01	5-10-1	41.91	6-10-1	40.17
4-10-1	41.02	5-11-1	40.98	6-11-1	39.62
4-11-1	40.61	5-12-1	39.61	6-12-1	39.03
4-12-1	40.51	5-13-1	39.52	6-13-1	40.28
4-13-1	41.61	5-14-1	39.63	6-14-1	40.85
4-14-1	42.08	5-15-1	39.94	6-15-1	41.91
4-15-1	41.16	5-16-1	40.17	6-16-1	40.42
4-16-1	42.04	5-17-1	40.60	6-17-1	40.49
4-17-1	42.97	5-18-1	41.16	6-18-1	40.85
4-18-1	42.33	5-19-1	40.86	6-19-1	41.61
7-7-1	41.52	8-7-1	40.63	9-8-1	41.34
7-8-1	40.78	8-8-1	39.09	9-9-1	39.53
7-9-1	39.61	8-9-1	38.72	9-10-1	39.15
7-10-1	38.93	8-10-1	37.59	9-11-1	38.33
7-11-1	38.06	8-11-1	36.42	9-12-1	37.88
7-12-1	39.18	8-12-1	35.84	9-13-1	37.62
7-13-1	39.38	8-14-1	36.88	9-14-1	35.92
7-14-1	40.94	8-15-1	36.11	9-15-1	34.51
7-15-1	41.38	8-16-1	37.66	9-16-1	35.92
7-16-1	41.62	8-17-1	38.03	9-17-1	35.60
7-17-1	42.88	8-18-1	39.82	9-18-1	37.82
7-18-1	41.02	8-19-1	38.86	9-19-1	38.91

Results for Detrended (Category 2) Data

Model	AARE	Model	AARE	Model	AARE
1-6-1	21.59	2-7-1	20.66	3-7-1	23.55
1-7-1	18.97	2-8-1	20.42	3-8-1	23.31
1-8-1	22.31	2-9-1	19.55	3-9-1	19.11
1-9-1	23.42	2-10-1	21.16	3-10-1	19.85
1-10-1	23.53	2-11-1	23.96	3-11-1	20.21
1-11-1	22.29	2-12-1	22.56	3-12-1	20.80
1-12-1	21.97	2-13-1	24.92	3-13-1	21.34
1-13-1	22.31	2-14-1	25.55	3-14-1	21.31
1-14-1	24.42	2-15-1	26.66	3-15-1	21.84
1-15-1	24.21	2-16-1	26.96	3-16-1	20.61
1-16-1	24.81	2-17-1	26.55	3-17-1	20.23
1-17-1	25.72	2-18-1	26.16	3-18-1	20.95
1-18-1	26.52	2-19-1	26.86	3-19-1	20.72
4-7-1	22.45	5-8-1	20.62	6-8-1	20.02
4-8-1	21.62	5-9-1	19.71	6-9-1	19.90
4-9-1	20.12	5-10-1	19.82	6-10-1	18.62
4-10-1	20.02	5-11-1	17.42	6-11-1	17.88
4-11-1	19.19	5-12-1	16.92	6-12-1	15.52
4-12-1	17.81	5-13-1	16.08	6-13-1	16.12
4-13-1	17.67	5-14-1	17.01	6-14-1	16.52
4-14-1	18.88	5-15-1	17.53	6-15-1	17.21
4-15-1	18.11	5-16-1	17.69	6-16-1	17.69
4-16-1	18.02	5-17-1	18.90	6-17-1	17.49
4-17-1	19.40	5-18-1	19.16	6-18-1	18.85
4-18-1	19.33	5-19-1	19.86	6-19-1	18.61
7-7-1	20.01	8-7-1	20.85	9-8-1	20.33
7-8-1	19.22	8-8-1	19.71	9-9-1	19.12
7-9-1	18.45	8-9-1	18.83	9-10-1	18.62
7-10-1	16.68	8-10-1	17.12	9-11-1	18.02
7-11-1	17.32	8-11-1	16.60	9-12-1	17.92
7-12-1	15.25	8-12-1	15.12	9-13-1	16.02
7-13-1	16.29	8-14-1	16.32	9-14-1	15.16
7-14-1	16.28	8-15-1	16.98	9-15-1	14.14
7-15-1	17.38	8-16-1	17.82	9-16-1	15.84
7-16-1	18.62	8-17-1	18.92	9-17-1	15.69
7-17-1	19.88	8-18-1	18.16	9-18-1	16.70
7-18-1	20.02	8-19-1	18.86	9-19-1	17.62

Results for Detrended – Deseasonalized (Category 3) Data

Model	AARE	Model	AARE	Model	AARE
1-6-1	13.29	2-7-1	12.96	3-7-1	13.45
1-7-1	10.97	2-8-1	12.92	3-8-1	13.34
1-8-1	12.31	2-9-1	12.55	3-9-1	12.61
1-9-1	14.42	2-10-1	13.16	3-10-1	12.85
1-10-1	13.53	2-11-1	13.86	3-11-1	12.61
1-11-1	13.29	2-12-1	13.96	3-12-1	12.85
1-12-1	13.97	2-13-1	14.92	3-13-1	12.34
1-13-1	13.31	2-14-1	15.55	3-14-1	12.11
1-14-1	14.42	2-15-1	16.16	3-15-1	12.85
1-15-1	14.53	2-16-1	16.86	3-16-1	12.61
1-16-1	14.31	2-17-1	16.55	3-17-1	12.23
1-17-1	15.42	2-18-1	17.16	3-18-1	11.85
1-18-1	16.53	2-19-1	17.86	3-19-1	11.61
4-7-1	12.80	5-8-1	11.68	6-8-1	11.62
4-8-1	11.65	5-9-1	11.21	6-9-1	10.82
4-9-1	11.32	5-10-1	10.02	6-10-1	9.62
4-10-1	10.92	5-11-1	9.82	6-11-1	7.88
4-11-1	9.91	5-12-1	9.62	6-12-1	7.62
4-12-1	9.62	5-13-1	8.88	6-13-1	8.12
4-13-1	10.01	5-14-1	8.99	6-14-1	8.62
4-14-1	9.88	5-15-1	9.23	6-15-1	9.21
4-15-1	10.04	5-16-1	9.62	6-16-1	9.62
4-16-1	11.12	5-17-1	8.92	6-17-1	9.45
4-17-1	11.42	5-18-1	9.16	6-18-1	9.85
4-18-1	11.53	5-19-1	9.86	6-19-1	9.61
7-7-1	12.02	8-7-1	11.65	9-8-1	14.63
7-8-1	11.62	8-8-1	10.90	9-9-1	10.12
7-9-1	10.23	8-9-1	9.23	9-10-1	9.62
7-10-1	9.68	8-10-1	8.82	9-11-1	8.12
7-11-1	6.32	8-11-1	6.60	9-12-1	7.62
7-12-1	5.62	8-12-1	4.62	9-13-1	6.12
7-13-1	6.21	8-14-1	6.72	9-14-1	5.26
7-14-1	5.88	8-15-1	5.88	9-15-1	4.14
7-15-1	6.31	8-16-1	5.62	9-16-1	5.14
7-16-1	9.62	8-17-1	6.92	9-17-1	5.62
7-17-1	9.88	8-18-1	6.16	9-18-1	6.10
7-18-1	10.02	8-19-1	6.86	9-19-1	7.62

Table 5.6 Results for Multi Hidden-Layer ANN Models

Model	AARE	TS_0.5	TS_1	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
9-1-12-1	1.228	7.407	15 926	43.519	65.0	87.037	95.185	97.497	98.121	98.148
9-1-13-1	1.140	6.852	15.926	44.074	65.0	87.037	95.00	97.407	98.121	98.148
9-1-14-1	1.121	9.815	15.926	39.815	64.249	87.593	96.111	97.963	98.154	98 248
9-1-15-1	1.083	9.444	15.926	42.778	64.815	87.222	95.556	97.963	98.154	98.348
9-1-16-1	1.048	9.630	17.407	42.407	65.185	87.222	95.185	97.778	98.245	98.148
9-2-11-1	1.109	8.889	17.963	42.407	65.926	87.407	95.37	97.963	98.154	98.248
9-2-13-1	1.107	7.963	14.630	44.239	65.557	87.222	94.817	97.407	97.952	98.048
9-3-11-1	1.119	8 719	17.113	42.107	65.386	86.107	94.17	97.813	98.014	98.248
9-3-13-1	1.107	7.163	15.130	44.159	65.246	87.212	94.725	97.427	97.152	97.663
9-4-8-1	1.208	6.667	15.926	44.259	65.926	86.852	95.0	97.778	97.651	98.663
9-4-10-1	1.104	8.148	14.815	44.815	65.185	87.407	94.817	97.407	97.524	98.348
9-5-7-1	1.187	8.519	15.926	42.222	64.259	87.407	95.741	97.963	98.021	98.148
9-5-10-1	1.107	5.00	7.963	27.407	52.222	86.481	96.663	97.963	98.201	98.148
9-6-7-1	1.116	9.630	18.148	41.481	65.926	87.222	95.373	97.963	98.121	98.648
9-6-9-1	1.008	8.889	18.148	43.519	65.556	87.222	95.185	97.778	98.121	98.648
9-2-12-1	1.107	7.222	15.556	45.370	66.111	87.407	94.815	98.178	98.425	98.858
9-3-12-1	1.117	7.212	15.150	45.129	66.091	86.127	94.415	97.963	98.013	98.248
9-3-14-1	1.107	7.623	13.321	24.915	65.782	87.801	94.712	97.524	98.013	98.248
9-4-9-1	1.183	8.33	17.245	41.289	62.098	85.370	94.815	97.978	98.013	98.048
9-4-11-1	1.033	7.778	14.672	42.778	65.625	87.192	95.153	97.407	97.672	98.348
9-5-9-1	1.181	9.444	17.788	42.778	52.827	87.026	95.672	97.963	98.021	98.148

Table 5.7 Results of Multi Hidden-Layer ANN Models

Model	AARE	TS_0.5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
9-7-7-1	1.165	9.815	18.33	42.037	65.556	87.222	95.37	97.963	98.121	98.148
9-7-10-1	1.017	10.0	18.148	42.037	65.556	87.222	95 37	97.963	98.121	98.148
9-8-4-1	1.511	8.630	12.22	36.852	60.62	87.354	96 745	97,963	98.121	98.548
9-8-7-1	1.013	10.37	17.963	43.148	65.627	87.354	95.733	97 963	98.324	98 748
9-9-3-1	1.234	7.778	15.926	11.111	65.0	87.846	95.263	97.725	98.420	98.488
9-9-7-1	1.039	10.556	18.704	43.333	65.27	86.645	95.55	97.674	98.121	98.488
9-10-2-1	1.607	8.815	12.222	43.333	60.00	87.643	96.836	97.648	98.240	98.633
9-10-4-1	1.57	8.44	8.704	37.256	53.889	85.833	96.22	97.625	98.121	98.644
9-11-2-1	1.197	8.259	10.741	30.00	57.771	87.084	96.746	97.908	98.420	98.488
9-11-4-1	1.114	7.037	13.519	35.253	63.287	86.734	94.63	97.472	97.527	97.66
9-12-2-1	1.050	4.815	12.222	42.712	60.00	87.643	96.647	97.663	98.240	98.448
9-12-5-1	1.015	6.852	15.556	37.263	66.725	87.222	96.784	97.552	98.240	98.448
9-13-2-1	1 081	6.852	13.519	45.627	59.736	82.723	94.783	97.52	98.241	98.333
9-13-4-1	1.041	7.407	14.815	42.526	65.836	87.222	94.734	97.627	98.121	98.418
9-14-2-1	1.099	6.481	12.407	45.0	60.556	86.354	94.815	97.772	97.963	98.117
9-14-4-1	1.010	7.778	14.815	39.259	63.265	86.685	95.837	97.562	97.973	98.148

Table 5.8 Results for Multi Hidden-Layer ANN Models

Model	AARE	TS_0.5	TS_1.0	TS_5	TS_10	TS_15	TS_25	TS_50	TS_75	TS_100
1-2-3-1	1.288	7.037	15.37	45 37	65 926	87.593	94.815	97.407	97.772	98.148
1-2-4-1	1.136	9.630	17.963	43.33	65.370	87.222	95.185	97.963	98.102	98.248
1-2-5-1	1.037	2.592	5 37	19 074	44.44	85.741	96.852	97.963	98.126	98.319
1-2-6-1	1.008	2.778	6.296	17.778	41.481	84.533	96.852	98.148	98.210	98.488
1-2-7-1	1.031	6.111	15.741	43.704	65.185	87.421	94.523	97.401	98.231	98.367
1-3-2-1	1.140	9.258	17.572	42.704	64.63	87.222	94.523	97.963	98.232	98.468
1-3-3-1	1.039	8.741	10.872	35.963	42.037	57.53	80.534	94.963	96.825	98.148
1-3-4-1	1.005	9.815	17.662	35.186	65.741	87.322	80.625	97.963	98.120	98.148
1-3-5-1	1.187	9.562	17.702	42.778	64.63	87.222	95.836	97.963	98.120	98.148
1-3-6-1	1.094	6.741	15.67	25.963	42.872	57.532	95.523	94.523	98.120	98.248
1-4-2-1	1.114	9.074	17.536	42.185	65.723	87.419	90.523	97.963	98.200	98.128
1-4-3-1	1.019	7.778	16.76	44.963	65.142	87.778	95.741	97.963	98.120	98.248
1-4-4-1	1.012	9.815	17.663	42.778	64.623	86.645	95.432	97.963	98.120	98.348
1-4-5-1	1.003	10.185	17.625	43.889	65.534	87.234	95.521	97.778	98.200	98.519
1-5-2-1	1.003	9.444	16.766	42.963	65.435	87.222	95.152	97.963	98.268	98.140
1-5-4-1	1.002	8.702	17.76	42.963	65.421	97.422	95.421	97.963	98.110	98.480
1-6-2-1	1.098	9.630	17.625	42.407	65.142	87.312	95.412	97.963	98.120	98.148
1-6-3-1	1.024	7.037	17.573	44.63	65.142	87.321	95.152	97.57	98.524	98.148
1-7-2-1	1.002	10.00	17.593	42.778	65.521	87.132	95.182	97.778	98.120	98.448